EXHIBIT 5

FOLEY

FOLEY & LARDNER LLP

ATTORNEYS AT LAW

555 SOUTH FLOWER STREET, SUITE 3500
LOS ANGELES, CA 90071-2411
213.972.4500 TEL
213.486.0065 FAX
WWW.FOLEY.COM

WRITER'S DIRECT LINE
213.972.4599
wrobinson@foley.com EMAIL

CLIENT/MATTER NUMBER
106884-9003

June 26, 2015

VIA E-MAIL

Austin Hansley, Esq.
Austin Hansley P.L.L.C.
5050 Quorum Dr., Suite 700
Dallas, TX 75254
Austin@TheTexasLawOffice.com

Re: Graham Springs LLC v. Provide Commerce, Inc., 2:15-cv-793 (E.D. Tex.)

Dear Mr. Hansley,

This firm represents Provide Commerce, Inc. ("PC") in the above-identified matter, and we have obtained a copy of the complaint. I have reviewed this matter with PC, and the fact of the matter is that PC should have never been sued. PC simply does not infringe any of the claims of the '085 patent. Furthermore, the '085 patent is clearly invalid in view of the substantial amount of prior art discussed below, and for claiming non-statutory subject matter according to the Supreme Court's recent *Alice* decision.

In view of the issues discussed below, it is PC's view that the complaint should never be served, and that if it has been served then this case should be dismissed with prejudice immediately. Pursuit of this action in view of what we provide below would not only constitute a violation of Rule 11, but also render the case exceptional under § 285, so as to entitle PC to its attorneys' fees.

A. Non-Infringement

PC simply does not infringe any claim of the '085 patent. Your complaint mentions the steps of claim 13 of the '085 patent. PC, however, does not practice any of the steps of claim 13 of the '085 patent. Claim 13 requires, among other steps, provding in a page description language file a data string operable of a multimedia device. PC does not provide any such page description language file. Moreover, because PC does not provide any such page description language file, it also does not perform the other steps of claim 13 of the '085 patent.

The lack of any facts in your complaint only underscores the lack of any infringement by PC, because the complaint does not mention any action performed by PC relating to the steps of the claims. We also note that the '085 patent had expired by Sep. 12, 2014, and the complaint makes no mention of any actions performed by PC prior to that date.

In view of the complete non-infringement of the claims, there can be no reasonable basis for your complaint against PC.

B. Invalidity Due to Prior Art

After reviewing the complaint, we examined the prior art and have discovered numerous references that clearly invalidate claim 13 of the '085 patent in view of how it appears you are trying to interpret that claim in the complaint. The extremely large number of prior art references that are all directly on point demonstrate the complete invalidity of the '085 patent. An even more stunning fact is the broad list of institutions associated with those prior art references, such as Hewlett-Packard, Adobe Systems, NeXT Computer Inc., and multiple research institutions. It seems that everyone all across the globe knew about such functionality and had already put it to use years before the filing date of the '085 patent. Thus, claim 13 of the '085 patent is clearly invalid in view of just a sampling of that prior art, as demonstrated below.

1. Williams et al., "Mail Facilities in a Multimedia Environment," Hewlett-Packard Journal, April 1994

The first reference we located is an article by Williams et al. titled "Mail Facilities in a Multimedia Environment" (hereinafter "the Williams article"). The Williams article was published in the Hewlett-Packard Journal in April 1994, which is well before the filing date of the '085 patent. The Williams article describes an actual working product of Hewlett-Packard, which was the HP MPower Mail System. A copy of the Williams article is attached to this letter as Exhibit A.

The Williams article discusses the use of a "MIME file" that includes both multimedia data and data for printing. (Williams article, pp. 72-73, 75-77). The content types that can be included in the MIME file are shown on page 76 of the Williams article, and include text, images, audio, and video. Williams explains that the several different types of multimedia data are contained in a single file in the MIME format, so that the single MIME file can include text, image data, audio data, and video data. *Id*. at pp. 75-76.

The multimedia data in the MIME file is used to operate a multimedia device. For example, an audio player program can be invoked to play the audio data from the MIME file. *Id*. at p. 72. Importantly, during printing of the MIME file, the audio data is excluded from being printed. *Id*. at p. 73. Williams describes the printing of the MIME file on page 73 as follows:

> "**Printing.** Multimedia printing involves decomposing the MIME formatted file into its parts and invoking the HP MPower print action (HP SharedPrint) on each of the parts. … Invoking HP SharedPrint for mail is handled by the shell script mmprint.
>
> Several types of multimedia cannot be printed (e.g., audio files). This is handled by a print action for the unprintable file that maps to the special action **NONE**." (Williams article, p. 73) (emphasis added).

The Williams article, therefore, describes a MIME file that includes both text data and audio data in a single file. The audio data from the MIME file is applied to an audio player to provide audio output. When the MIME file is printed, the MIME file is decomposed into its

parts and the print action is invoked on each of the parts, but the multimedia data that cannot be printed (such as the audio data) is excluded from the printing operation. *Id*. at pp. 72-73. The Williams article demonstrates that such operations were known and implemented well before the filing date of the '085 patent.

2. Adobe Systems Inc., "The Future of PostScript Language Technology from Adobe Systems," September 1991

The second reference we located is a paper from Adobe Systems titled "The Future of PostScript Language Technology from Adobe Systems" (hereinafter "the Adobe paper"). The Adobe paper was published in September 1991, which is years before the filing date of the '085 patent. A copy of the Adobe paper is attached to this letter as Exhibit B.

The Adobe paper describes the use of PostScript files and, importantly, includes a section on page 6 titled "The PostScript Language as a Carrier of Multi-Media Information." (Adobe paper, p. 6). The Adobe paper explains that, as of September 1991, the PostScript language could already carry audio information, which allows users to add voice messages to documents. Indeed, on page 6, the Adobe paper states the following:

> "The PostScript language can already carry audio information, as demonstrated by applications for NeXT computers, which allow users to add voice messages to documents." (Adobe paper, p. 6) (emphasis added).

We have obtained textbooks and programs for "NeXT computers" from the early 1990's, which show that the Display PostScript extensions developed by Adobe Systems for NeXT computers included a "playsound" PostScript instruction for playing audio, and also included extensions for controlling a video monitor. As the Adobe paper explains, "[a] 'document' in the PostScript language is therefore an electronic file containing any or all of the following elements: text, line art, scanned or sampled images (photographs), sound and full-motion video." *Id*. at p. 6 (emphasis added). The sound is played while the text portion of the document can be printed. *Id*. The Adobe paper demonstrates that such concepts were known and implemented well before the filing date of the '085 patent.

3. Isle et al., "Multimedia Interface and Method for Computer System," U.S. Patent No. 4,931,950, Issued June 5, 1990

The third reference we located is U.S. Patent No. 4,931,950 by Isle et al. titled "Multimedia Interface and Method for Computer System" (hereinafter "the Isle patent"). The Isle patent was issued on June 5, 1990, which is years before the filing date of the '085 patent. A copy of the Isle patent is attached to this letter as Exhibit C.

The software described in the Isle patent interprets commands in files that specify audio messages, text to be printed, and video sequences to be played. (Isle, col. 5, line 34 – col. 9, line 68). The commands in Isle are specified by "@" command markers and, for example, the symbol "@s" is a command to speak text using audio output, and the text is spoken up to a

closing "@" symbol in the file. *Id*. at col. 6, lines 3-14; col. 7, line 31 – col. 8, line 68; Table 1. Similarly, the command "@p" in the file specifies text to be printed, and continues up to a closing "@" symbol. *Id*. at col. 8, lines 5-6; Table 1.

Thus, when a file is opened in the system of the Isle patent, the text between "@s … @" is synthesized and output through a speaker, while the text between "@p … @" is printed by a printer. *Id*. at col. 5, line 34 – col. 9, line 68; Table 1. In other words, the data string between "@s … @" in the file is used to operate the synthesizer and speaker by being applied to control the synthesizer and speaker, while the portion of the data that excludes that spoken text and is between "@p … @" in the file is printed. *Id*. The Isle patent demonstrates that such operations were known and implemented well before the filing date of the '085 patent.

C. Invalidity Due to Non-Statutory Subject Matter

The Supreme Court's recent decision in *Alice Corp. v. CLS Bank*, 134 S. Ct. 2347 (2014), reaffirmed that patent claims are statutorily invalid under 35 U.S.C. § 101 when they recite nothing more than abstract ideas. The Supreme Court emphasized that claims that merely require generic computer implementations fail to transform an abstract idea into a patent-eligible invention. The claims of the '085 patent are even more deficient than the claims that were at issue in *Alice* and, therefore, are invalid.

Moreover, courts are not waiting until summary judgment motions are filed to invalidate patents under § 101. We urge you to review, for example, the CAFC's September 3, 2014 opinion in *buySAFE, Inc. v. Google, Inc*., 765 F.3d 1350 (Fed. Cir. 2014). This decision denied patent-eligibility under § 101. While that result is no surprise in view of *Alice*, the vehicle for the decision in the district court was a motion for judgment on the pleadings rather than a summary judgment motion. No claim construction was necessary.

The claims of the '085 patent relate to nothing more than the abstract idea of providing and applying data. Mankind has been providing and applying data since the beginning of time. The Court in *Alice* addressed the argument that 'the introduction of a computer into the claims rendered patentable an invention that otherwise would not have been." 134 S. Ct. at 2357. Citing *Gottschalk v. Benson*, 409 U.S. 17 at 67, the *Alice* Court rejected that argument on the ground that computer implementation of an abstract idea "did not supply the necessary inventive concept; the process could be carried out in existing computers long in use." 134 S. Ct. at 2357.

Claim 13 of the '085 patent is even more abstract than the claims at issue in *Alice*, because it does not even mention a computer to implement what is clearly an abstract idea. The '085 patent claims are thus clearly invalid under § 101 and Graham Springs should immediately drop its complaint in view of that invalidity. The era of software patents, and the trolls that assert them, is over. If you proceed to serve the complaint, PC will consider all options, including proceeding with a Rule 12(b)(6) motion to invalidate the patents under § 101 and then following that with a § 285 request for fees.

D. Violation of Rule 11

In view of the above, the serving of the complaint would be a violation of Rule 11 because the complaint is objectively baseless and, therefore, serving the complaint would justify the imposition of sanctions against Plaintiff. *Raylon, LLC v. Complus Data Innovations, Inc.*, 700 F.3d 1361 (Fed. Cir. 2012). As the concurring opinion in *Raylon* notes at 1375:

> Indeed, [defendant] memorialized its concerns by writing to [plaintiff] early in the case regarding the merits of its allegations, but it appears that [plaintiff] took no curative measures and proceeded through claim construction undeterred. J.A. 3139 ("As we have repeatedly told you since day one, these infringement allegations are violations of Rule 11."); J.A. 3141-43; J.A. 3145-46; J.A. 3153-54 (counsel correspondence discussing perceived violations of Rule 11). This early notice at least suggests that [plaintiff] had a duty to "stop and think," and to investigate whether its positions were objectively baseless, but it opted to ignore grappling with the incongruence raised by visual comparison of the accused devices with the plain language of the claims. (emphasis added).

As was done for the plaintiff in *Raylon*, we provide this early notice to you so that you can fulfill your duty to "stop and think." When you do, you will realize that the patent is completely invalid and not infringed, rendering the complaint utterly without merit. If the complaint is served, PC will consider its options, including seeking costs and fees against plaintiff and its counsel for bringing and maintaining a frivolous case.

I look forward to hearing from you.

Very truly yours,

FOLEY & LARDNER LLP

William J. Robinson

EXHIBIT A

Mail Facilities in a Multimedia Environment

Providing a multimedia email facility required that the well-established processes of creating, sending, receiving, printing, and replying to email messages be maintained and applied to messages containing multimedia objects.

by Robert B. Williams, Harry K. Phinney, and Kenneth L. Steege

The advent of tools and capabilities that allow users to manipulate and create multimedia objects on a workstation mandated the need to make it possible to send these objects through electronic mail, or email. The user interface for creating, reading, and sending text messages through email is well-established. For multimedia email to be effective the same sort of process flow must be in place. For example, just as a user can use the more command to view a standard text email message, an equivalent facility must be available to view a multimedia email message. What this implies is that the user should not have to be concerned with invoking the correct software to deal with a particular media type because this should be handled by the mail facility.

The HP MPower mail facility, which is represented by the envelope icon on the HP MPower front panel, provides support for sending, replying, viewing, and printing of multimedia mail.

For sending multimedia email, HP MPower provides two approaches: dragging and dropping a file on the envelope icon or clicking on the envelope icon. With the drag-and-drop method, the user is presented with the dialog box shown in Fig. 1a. From this box the user can either send the dropped file to its destination by selecting the Send button or edit the file by selecting the Edit button. If the Edit button is selected, the mail composer (editor) window is displayed showing the contents of the file that was dropped on the mail icon (see Fig. 1b).

To read mail the user can click on the mail icon and be presented with the two windows shown in Fig. 2. The first window contains the standard elm screen and the other contains the HP MPower viewer screen. Elm, which is the HP-UX* screen-oriented electronic mail processing system, provides the user interface for the user to interact with the HP MPower mail system. To edit or create a multimedia mail message, the user can access the HP MPower mail editor (composer) by selecting Mail Msg from the elm screen. This will provide the screen shown in Fig. 1b.

To read or view a mail message, the user would select one of the messages in the message list and then press Read Msg menu item from the elm screen shown in Fig. 2. The selected mail message will appear in the HP MPower viewer window for reading.

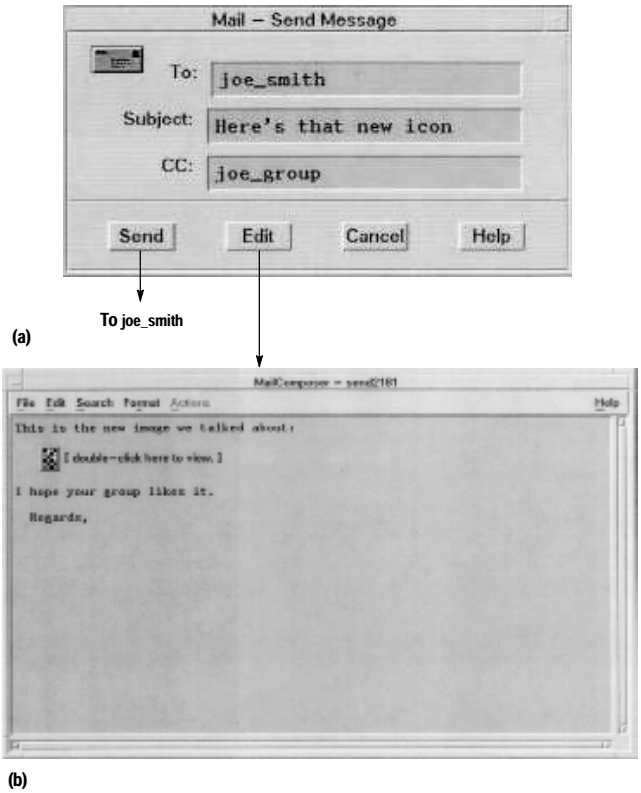


Fig. 1. (a) The dialog box that appears after a mail message is dragged and dropped on the front-panel mail icon. Selecting the Send button will send the message to its destination. (b) Selecting the Edit button produces the mail composer screen.

HP MPower Mail System Components

The main components of HP MPower mail include the HP-UX elm mail user agent, a multimedia editor, several shell scripts, a standard multimedia file format and supporting software, and HP VUE actions and file types. Fig. 3 shows a simplified diagram of some of the main HP MPower mail components responsible for providing the user interface actions described above.

With the exception of vuemime, which among other things handles the encoding and decoding of multimedia data, the

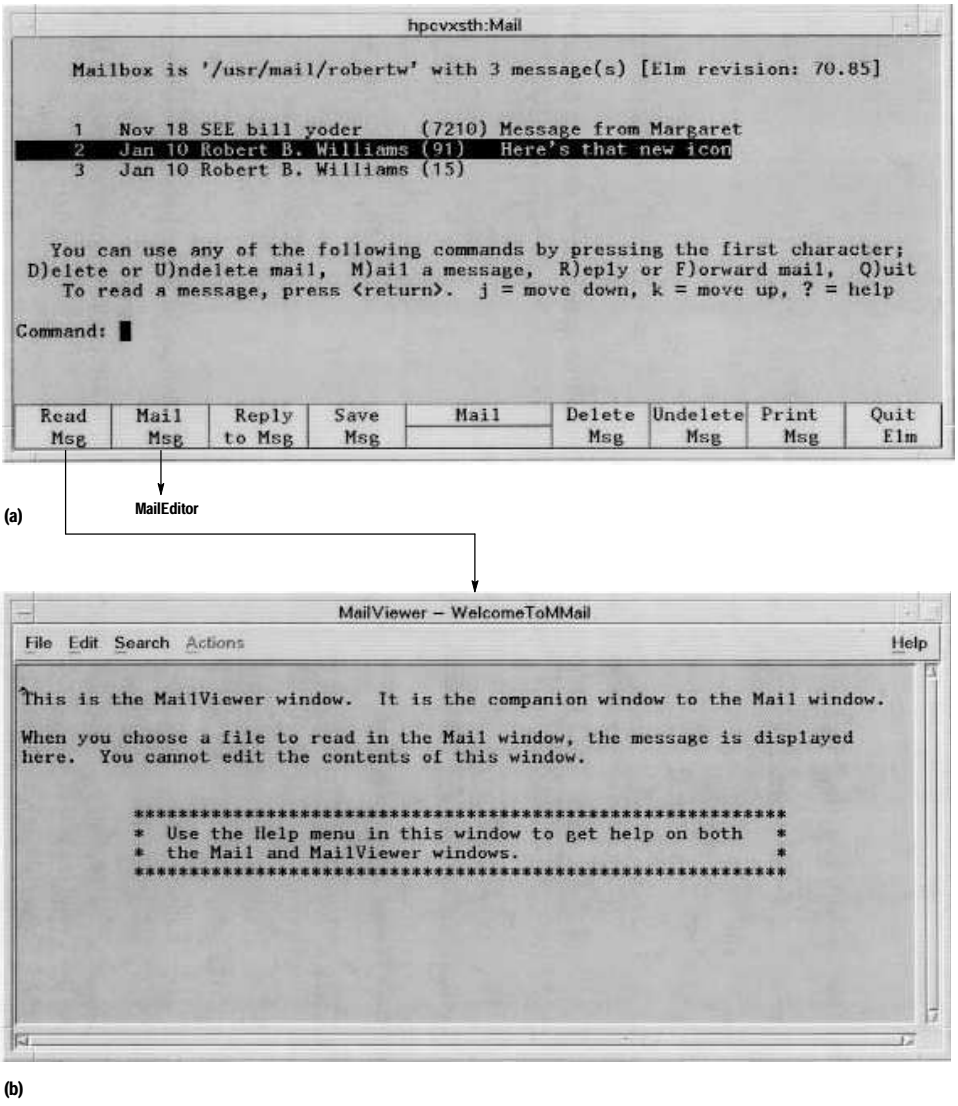


Fig. 2. The windows produced by just clicking the mail icon. The top window contains elm and the lower window contains the HP MPower viewer.

components shown in Fig. 3 that send or receive data require no specific knowledge about dealing with multimedia data. For example, when a media-rich file is sent to sendmail, the file is treated like any other file going onto the network. Another example is when the composer or viewer needs to render a multimedia object (e.g., play an audio file or draw an image), the media icon is mapped to the process (or actions) capable of handling the particular media object by the HP VUE action database.

Vuepad. This is an enhanced version of the HP VUE editor. It provides two modes of operation in the HP MPower mail system: composing (creating a multimedia mail message to send) and viewing (reading a multimedia mail message). Vuepad is described in detail later in this article.

HP VUE Action Database. When a particular file type is selected the HP VUE action database provides the mechanism for invoking the appropriate action associated with that file type. For example, when the user double-clicks on an icon representing an audio file, the HP VUE action database tells vuepad that the audio player should be invoked to play the contents of the file.

MIME, Vuemime, and Metamail. To handle the interchange and storage of different types of data, HP MPower uses an internet standard known as MIME (Multipurpose Internet Mail Extensions). MIME is an extension to the basic internet mail standard RFC 822, which specifies the format for internet text messages. MIME defines the format for multipart, multimedia mail messages. Vuemime and metamail are utilities that provide support to HP MPower for MIME data. Vuemime provides a single interface for HP MPower components using MIME data, and metamail is a public-domain utility that we modified to act as a MIME filter. In this role metamail decodes and flattens MIME messages from non-HP MPower sources and encodes messages going to the mail transport agent sendmail. Vuemime is the only HP MPower component that communicates directly with metamail. MIME, vuemime, and metamail are described in more detail later in this article.

Mail User Agents. HP MPower has two mail user agents, elm and a shell script called mmsend. Elm is the main user agent, providing the usual elm capabilities of viewing, printing, saving, or deleting incoming or stored mail, replying or forwarding mail, and originating mail. Elm is the standard mail

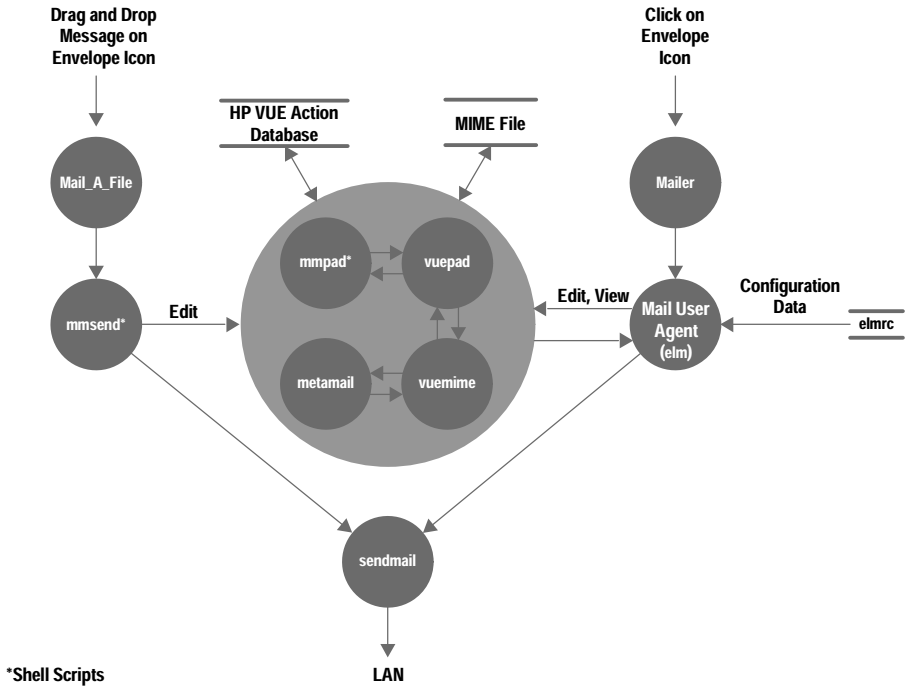


Fig. 3. Some of the main components contained in the HP MPower mail facility.

processor that is shipped with every HP workstation. This greatly reduces the support load and avoids duplication of development effort. Unfortunately it also put us in the position of offering a keyboard-based mail agent amidst a large collection of applications with graphical interfaces. Access to elm is through clicking on the mail icon, producing the screens shown in Fig. 2.

One of the obvious changes to elm's normal interaction model is the tools used to view and edit mail. In the HP MPower environment, since we have to deal with mail that may contain nontextual items, the multimedia composer (editor) and viewer in vuepad are used to edit and view mail. This is done by modifying the user's elmrc (configuration) file so that instead of invoking vi or more for editing or viewing a file, HP MPower's editor or viewer are invoked. Most other user customizations of elm behavior in the elmrc file are preserved.

The second mail user agent, mmsend, provides an interface for drag and drop mailing of multimedia objects. Dropping a file or files on the mail icon causes the software to invoke an HP VUE action that indirectly calls the mmsend shell script. Mmsend drives the processes that are responsible for displaying and interacting with the screens shown in Fig. 1. The mmsend script also invokes the processes that convert multimedia files into MIME format. When the user selects the Send button from the mail dialog box shown in Fig. 1, mmsend invokes the mail transfer agent sendmail to dispatch the message over the network.

Mail Transfer Agent. The primary mail transfer agent in the current version of HP MPower is sendmail, the HP-UX facility for sending mail over the internet. Although there are gateways from sendmail SMTP (Simple Mail Transport Protocol) to X.400 and OpenMail, these gateways are not supported because they have not been enhanced to understand MIME on the SMTP (outbound) side.

Instant ignition systems (see page 17) usually do not have sendmail enabled. To overcome this the HP MPower installation scripts set up a default sendmail.fc and alias database. This is only done if sendmail is not already enabled. It does not address sendmail connectivity issues in complex or restricted environments. Ideally, sendmail would be turned off on mini-client systems because the mail agent runs on the server with the rest of HP VUE.

Printing. Multimedia printing involves decomposing the MIME formatted file into its parts and invoking the HP MPower print action (HP SharedPrint) on each of the parts. HP SharedPrint is described in the article on page 44. Invoking HP SharedPrint for mail is handled by the shell script mmprint.

Several types of multimedia cannot be printed (e.g., audio files). This is handled by a print action for the unprintable file that maps to the special action NONE.

Since printing actions happen on the client (HP SharedPrint forces this), vuemime and its database mimetypes must exist on the client as well as the server. This has implications for installations adding multimedia types.

Multimedia Editor

An important aspect of a multimedia mail facility is the ability to compose and view documents containing both text and nontextual items. In the HP MPower environment this ability is provided by a specially evolved version of the HP VUE text editor, vuepad.† All of the multimedia editing and viewing facilities available in HP MPower are built on this new version of vuepad.

The development of this component was constrained by many different factors. The schedule was set to allow the

† Unless stated otherwise references to vuepad in the rest of this article will be referring to the new version of the program.

introduction of the product to coincide with the release of new workstation computers. This ensured good public exposure for the product, but limited the amount of time and the number of engineers available for product development. The schedule and staffing level provided a strong motivation for using the functionality of existing products rather than wholesale development of new components.

Vuepad Modes. The vuepad program provides two modes. One is the HP MPower mail composer (mentioned previously) for creating and editing multimedia documents. The other mode is the HP MPower mail viewer (also mentioned previously) for viewing or listening to multimedia documents. The viewer is simply a read-only version of the composer. These facilities are accessible from the HP MPower mail icon or the HP VUE panel edit icon. The activation of either of these two modes is determined by the arguments passed to the vuepad program when it is invoked from the mmpad script shown in Fig. 3.

The HP MPower mail composer takes files created by the various media editing tools such as the HP SharedX Whiteboard or the audio editor and allows the user to incorporate the files generated by these tools into a document containing text. The final output of a message created by the composer is in a format compatible with the MIME standard. This approach required no modifications to any of the media editing tools and no changes to the core email application (elm). Use of the MIME message format helps provide some interoperability with other mail systems and gives us access to an already well-documented and carefully defined structure for messages.

Fig. 1 shows that the vuepad editor provides the composer and viewer with a visually appealing "iconic" view of nontextual data. The vuepad editor ensures that a user's normal editing actions function as the user expects in the presence of nontextual items. It also ensures that the operations available for manipulating nontextual items are clearly visible.

Modifications to Vuepad. The required modifications to the old vuepad editor included adding the ability to invoke a filtering program for reading and writing multimedia data. The choice to do this filtering in a separate program was primarily driven by the need to develop the filtering functionality in parallel with the enhanced editor, and a desire to use different development programming languages for the editor and filter.

The filter is the vuemime program mentioned earlier. When vuepad is reading a multimedia file, vuemime strips out the media data, writing each media object to a separate temporary file. The media data is replaced within the original data stream by a special character sequence indicating that a media object existed in that location. This character sequence, known as a tag, contains the name of the temporary file that holds the media data. This file is used by vuepad and the HP VUE file-typing facilities to determine the graphical icon to display in place of the media object. When vuepad writes to a multimedia file, vuemime inspects the file for any tags and replaces the tags with the actual media data and the necessary MIME header information.

It was necessary to augment or override some of the internal functions of the OSF/Motif text widget that vuepad is based on. Before the widget draws a line of text, vuepad checks to see if the text corresponds to part of a media icon. If it does, then vuepad draws the necessary portion of the icon. If the text does not contain part of an icon, then the text widget is allowed to render the whole line of text. Whenever the widget writes or reads data to or from either the OSF/Motif clipboard or the X primary selection window (a clipboard that can hold one item at a time), vuepad runs the data through the vuemime filter program to reinsert or strip any embedded media data as required. When the user selects a portion of text for an editing operation, vuepad tracks the selected region to provide meaningful highlighting of any selected media icons. Vuepad also observes all deletions and additions to the document to accurately track the position of the media objects within the document. Special care is also taken during the spell checking and formatting operations. The spell checking code must exclude the rather cryptic media tag data from the text sent to the spell checking program, and the formatting code has to do its work while preserving the relative locations of the media icons.

Compatibility. The behavior of vuepad is controlled by X Window System resources and command line options that allow vuepad to behave identically to the previous nonmedia-enhanced version of the HP VUE editor. The HP VUE file typing† and action database mechanisms were used to speed development and to provide consistency with HP VUE's file manager appearance. The HP VUE action database provides a simple means for invoking an appropriate action associated with any particular file type. Use of the action database in vuepad ensures consistency with the default action accessed from the file manager for a particular file type.

Composer Features. The HP MPower multimedia composer presents a flat, scrollable view of a document. This allows the user to access any and all document parts rapidly with no enforced sequential ordering. The media objects are embedded in the document, that is, the actual media data is copied into the resulting file rather than having links maintained to the original data. This ensures that the recipient of a message not only has immediate access to the data, but also results in larger messages and freezes the data at the time of composition. The media objects can be embedded at any location within the document, providing significantly more flexibility than the "attachment" model used by many mailers. The attachment model maintains links to other parts of a document.

To incorporate a media object into a document being composed, the user can either use the Include dialog facility of vuepad or drag the object from a file manager view and drop it on the composer's window. If the user wishes to create a new media object while composing a message, the appropriate editing tool (e.g., the audio editor or the image editor) must be used to create the object within that editor. After the media object is created and saved in the file system, vuepad's Include dialog or drag and drop facilities can be used to include the new object in the message being composed.

As shown in Fig. 1, the composer displays icons in place of all nontextual portions of the document. This results in a reasonably attractive appearance and good user recognition because of the similarity with the HP VUE file manager

† The HP VUE file typing mechanism provides the capability to define classes of files. For example, it can define all files ending with .tif to be of class TIFF.

iconic view. The user can either double-click with the mouse, or press the **Return** key to activate the currently selected icon. This provides a simple and quick means of viewing or playing the media item and allows easy access from the keyboard for those users who prefer not to move their hands to the mouse.

The composer also provides an **Actions** menu item which is sensitive whenever a single media item is selected. This menu contains two items. The first item is **Open**, which duplicates the functionality of double-clicking on the selected icon. The second item is **Save As**, which allows the user to save the selected media object to another file separately, with no MIME structure. The **Save As** action is also the default "open" action of the **unknown** data type. The **unknown** data type allows users to pass arbitrary binary data through the mail system undisturbed.

The current implementation of the composer does have some limitations, but a foundation has been built upon which to improve and add features to the current version.

Multimedia Data Extensions

As mentioned previously, HP MPower supports the interchange and storage of several different types of multimedia data contained in a single message or a file via a format conforming to an extension to internet mail known as MIME (Multipurpose Internet Mail Extensions). This multipart, multimedia support is implemented in HP MPower by two utilities: **metamail** and **vuemime**. **Metamail** is a public-domain, sample implementation of a full-featured MIME agent. **Vuemime** is a MIME filter that was created specifically for HP MPower and HP VUE to simplify generation and manipulation of multipart, multimedia data from the various HP MPower components.

We selected MIME for its strong support within the internet mail community, and we found it to be also useful outside the mail domain. The HP MPower environment recognizes files structured in compliance with the MIME specifications and will deal with them transparently for editing and printing. The composer allows the user to insert nontextual objects into any document being edited, and the resulting file will be saved in MIME format. The MIME message format is also used for cut and paste operations between editing windows. This allows the user to treat editing of mixed-media data in the same manner as plain text. Unfortunately, the mail composer is currently the only application that understands MIME data in cut and paste operations.

This section first provides some background on MIME, describes **metamail** in the context of HP MPower, and then discusses **vuemime** in some detail.

MIME Background. Since 1982 the standards that form the basis for internet mail have been defined by RFC 822 and the SMTP (Simple Mail Transport Protocol) defined by RFC 821.† RFC 822 was intended to specify a format for text messages and, as such, did not explicitly allow for inclusion of multimedia messages such as audio or image data. In particular, RFC 822 defines a message as consisting of two parts: a header and a body. The header consists of a series of specific field names and field values followed by a blank line that marks the end of the header and the beginning of the body. The body is restricted to relatively short lines (1000 characters) of seven-bit ASCII characters which cannot exceed a certain length. Users who wish to include nontextual data have to convert the data to seven-bit ASCII before submitting the data to their mail user agent or mail program.

RFC 1049 attempted to rectify some of the deficiencies of RFC 822 by defining a header field and a content type that marks the entire message body as being a certain type of data (e.g., text, audio, video, etc.). In the absence of a content-type field, the body was assumed to be U.S. ASCII text, as before. Although RFC 1049 has been used by several implementations, it is not without problems. The most severe problem is its total lack of support for multipart mail. RFC 1049 allows a message body to be specified as containing something other than text, but only one such thing.

RFC 1341, or MIME, generalizes and extends RFC 1049 in several ways. Most important, it defines a new content type called *multipart,* which can be used to encapsulate several body parts within a single RFC 822 message body. It also goes far beyond RFC 1049 in explicitly describing the set of allowable content types by defining a subtype mechanism for content types that includes provisions for addressing standardized encoding of non-ASCII character sets. It should be noted that RFC 1341 is an extension to, rather than a revision of RFC 822 in that it defines these new features (including text of unlimited line and overall length, characters sets other than ASCII, and multifont messages) within the confines of RFC 822.

The new header fields and content types defined in the MIME extension are described on the next page.

Metamail. **Metamail** is a public-domain, sample implementation of a MIME agent that was designed to function as a back end for an existing mail user agent. It can be incorporated into virtually any mail reading (or bulletin board) program (e.g., **xmail**, **xmh**, **elm**, etc.), enabling it to become a multimedia reading interface. **Metamail** knows how to parse a structured MIME message, flatten any hierarchy of nested messages, decode the various parts, and optionally, dispatch the appropriate handlers or viewers for the different parts. The commands used to dispatch the handler or viewer for each content type are specified in one or more **mailcap** (configuration) files, which allow a great deal of flexibility in adding and configuring handlers. As a viewer, **metamail** can be thought of as a multimedia counterpart to the ASCII paging tool, **more**, except that it enlists the aid of additional handlers and viewers when paging through a message in a linear fashion.

In HP MPower we needed more flexibility when composing, viewing, printing, and sending mail so we did not incorporate **metamail** directly in our mail user agent (**elm**). Instead, we use the enhanced version of **vuepad** described above as our mail viewer and composer and we delegate **metamail** to the role of a MIME filter or preprocessor and postprocessor. In this role, **metamail** serves primarily to simplify the digestion and generation of MIME compliant messages by other HP MPower components. Specifically, **metamail** is used to flatten potential message hierarchies and to encode and decode each message part according to its encoding scheme. On the input side (mail receiving), for example, if we receive a MIME message from a non-HP MPower sender, we pass it to

† Each internet standard is defined by one or more standards each known as a Request for Comment, or RFC.

MIME Header Fields

The MIME extension (RF 1341) to the basic internet mail standard RFC 822 created the following new header fields:

- MIME Version. This field is used to specify a version number that declares a message conformant with the MIME standard
- Content Type. This field is used to specify the type and subtype of data in the body of a message and to specify the complete encoding of such data
- Content Transfer Encoding. This field is used to specify auxiliary encoding applied to data to allow it to pass through mail transports having data or character set limitations
- Content Identifier and Content Description. These fields are used to further describe data in the body of the message.

Content Types

MIME enumerates precisely seven valid content types and requires that any additions to this set be specified in a new, similarly formal document. This restriction is a major change from RFC 1049, which allowed for much freer definition of new content types. Instead, the new mechanism for extensions is to define new subtypes of established content types. In general, implementors are required to register new subtypes with the Internet Assigned Numbers Authority (IANA) to avoid name conflicts. (An exception is private subtypes beginning with the letter X, which can be used freely and without registration.)

The seven defined content type values are:

- Text. This is the default content type. The default subtype is plain text. This content type has a charset attribute that has the default value us-ascii.
- Image. This content type is for still images. Subtypes are image format names (e.g., image/gif and image/jpeg).
- Audio. This content type is for audio information. Subtypes are audio format names. For example, audio/basic denotes single channel 8000-Hz μ-law audio data.
- Video. This content type is for video frames. Subtypes correspond to video format names such as video/mpeg.
- Message. This content type is used to encapsulate an entire RFC 822 format message. For example, it can be used in forwarding or rejecting mail. The standard defines two subtypes of message: message/partial, which can be used to break a large message into several pieces for transport so that they can be put back together automatically on the other end, and message/external-body, which can be used to pass a very large message body by reference, rather than including its entire contents.

  It should be noted that a message with a message content type can contain a message that has its own, different content-type field, meaning that the message structure can be recursive.
- Multipart. This content type is used to pack several parts, of possibly differing types and subtypes, into a single RFC 822 message body. The content-type field specifying a multipart type also includes a delimiter, which is used to separate each consecutive body part. Each body part is itself structured more or less as an RFC 822 message in miniature, possibly containing its own content-type field to describe its type. Subtypes of multipart types are specifically required to have the same syntax as the basic multipart type, guaranteeing that all implementations can successfully break a multipart message into its component parts. An expected use of multipart subtypes is to add further structure to the parts and to permit a more integrated structure of multipart messages among cooperating user agents.
- Application. This content type is for most other kinds of data that do not fit into any of the above categories, such as list servers, mail-based information servers, and PostScript.™

A separate part of the content-type header field can be used to convey supplemental information that may be either optional or required, depending on the content type. Such parameters are given in keyword = value format, and are used, for example, to convey information about character sets for text objects. Thus, the default message type for internet mail can be given a MIME content type of:

Content-type: text/plain; charset=us-ascii

Content Transfer Encoding

If internet mail transport (SMTP, as described by RFC 821) is ever upgraded to permit arbitrary binary data of unlimited line length in message bodies, the issue of encoding a message for transport will go away. However, even those who advocate such changes to SMTP generally recognize that they will be slow in coming. In the interim, there is wide perception that a standard mechanism for encoding arbitrary binary data for mail transport is needed.

The content transfer encoding header field can be used to specify the encoding technique used to render binary data in short lines of seven-bit data. After much debate, the working group settled on two encodings, which may be used interchangeably. One of them, the base-64 encoding, encodes each three bytes of binary data as four bytes of 7-bit data, using a base-64 alphabet selected for maximum portability across SMTP implementations, including ASCII to EBCDIC gateways. The other encoding scheme, quoted-printable, is a less efficient representation that preserves nearly all 7-bit ASCII characters as themselves. It is expected that base-64 will be preferred for genuine binary data, while quoted-printable will be preferred for data that is largely U.S. ASCII, but has scattered non-ASCII characters within it. In particular, this may be the preferred encoding for textual email in the national-use variants of ASCII, ISO 8859-X.

If the content transfer encoding field appears in the RFC 822 message header, it refers to the body of the message. If it appears in the header area of one part of a multipart message, it refers to the body area of that part only. The content transfer encoding field is prohibited when the content-type field has a value of multipart or message. This is necessary to prevent nested encodings.

metamail, which filters it by flattening any nested message hierarchies and decoding each message part. The result is a simplified MIME template that can be easily dealt with by other HP MPower components. On the output side (mail sending) metamail is used mainly to encode message parts for handling by the SMTP transport used by our mail transport agent sendmail. While we can handle nested messages on the input side, we only generate flat, single-level messages on the output side.

The official metamail documentation and software, including a draft of RFC 1341, can be found in the pub/nsb directory on thumper.bellcore.com.

Vuemime. In HP MPower, interaction with MIME messages is not the sole domain of the mail user agent (elm) and its viewer and composer (vuepad). Since MIME is used as the storage format for all multipart, multimedia data, printing and sending MIME data as well as composition and viewing have to be performed independently of elm. Printing and sending, for example, can be done by simply dropping a MIME file on the HP VUE front-panel printer or mail icons without invoking elm or vuepad.

Vuemime was created specifically for HP MPower components to provide a single interface to MIME data and, as such, vuemime is the only HP MPower component that directly enlists the services of metamail. Essentially vuemime can be viewed as a higher-level MIME filter which, in addition to providing the type of filtering performed by metamail, provides intermediate formatting that is easily digested by various HP MPower components. What has resulted, after analysis of the needs of various HP MPower components, is essentially five levels of formatted data which can be viewed as stages in a file's morphosis from raw data (level 0) into a fully formatted

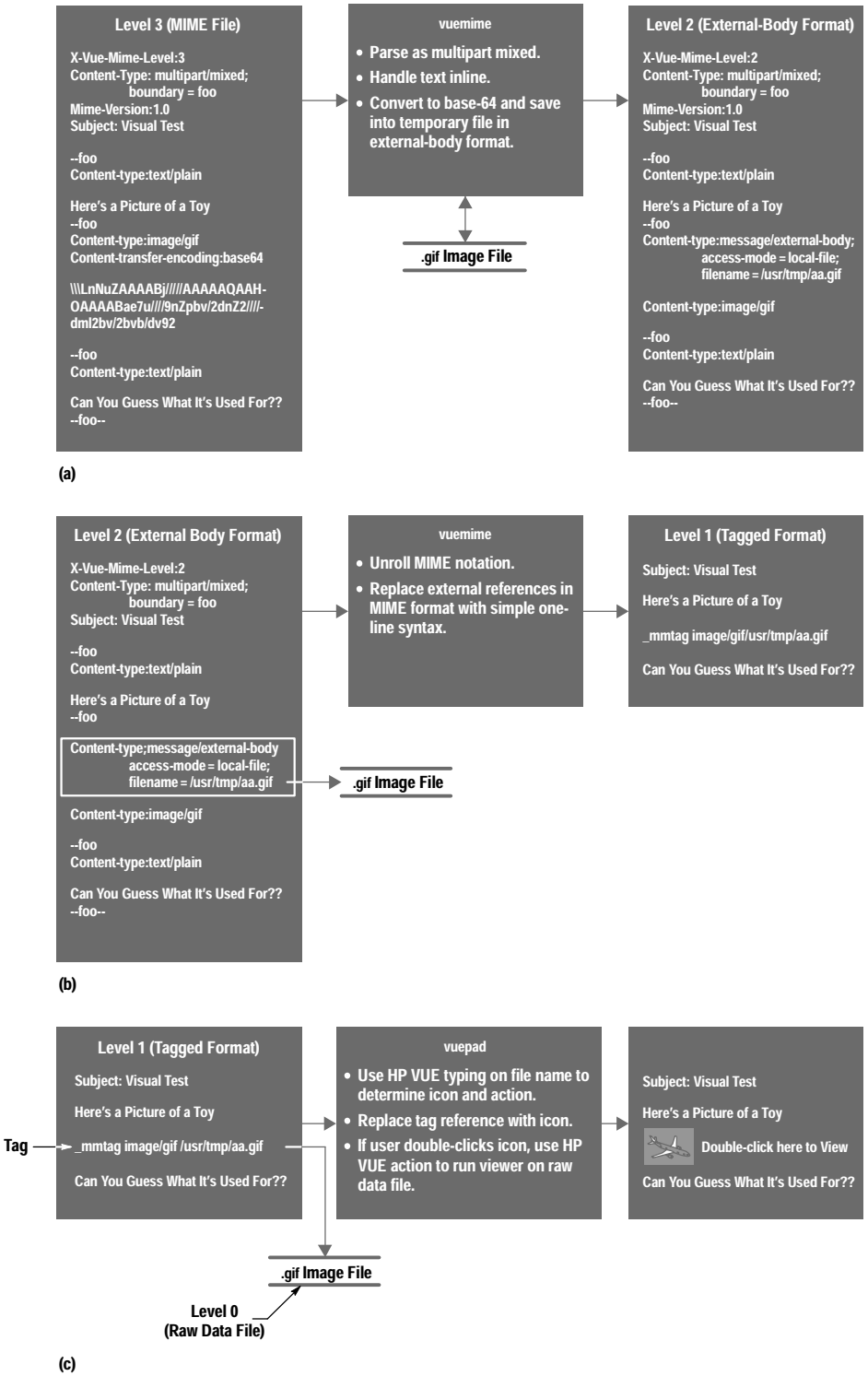


Fig. 4. Processing an incoming message. (a) Level-3 file format and the actions of vuemime to move from a MIME format to a level-2 file format. (b) Level-2 file format and the actions performed by vuemime to transform the data to a level-1 representation. (c) Level-1 file format representation and the actions taken by vuepad to display textual and nontextual parts of the message.

MIME message ready to be handed off to the SMTP mail transport.

Fig. 4 shows the steps involved in transforming an incoming message in MIME format (level 3) through the various levels of formatted data to a tagged (level 1) format. For an outgoing message, vuemime will take a tagged message and generate a self-contained MIME message that can be passed to the mail transfer agent.

The five levels of formatted data shown in Fig. 4 are defined as follows:

- Level 0. This is raw data consisting of ASCII text and non-textual information such as image, audio, and video data.
- Level 1. Data at this level, which is used by vuepad and the mmprint action, is in a special tagged format that contains only ASCII text, summarized mail heading information, and special tag lines representing multimedia objects. The tag lines indicate the multimedia type and a path to a file containing the raw data. See the level-1 file representation in Fig. 4c.
- Level 2. This is an external-body file format. The format at this level is very similar to level 1 except that MIME header and control lines have been added to make it fully MIME

compliant. All multimedia objects are still external (and raw) but are now identified by Content-type: <message>/<external-body> and Content-type: <type>/<subtype> lines. See the level-2 file representation in Fig. 4b.

HP MPower does not currently send messages in level-2 file format, but this level could be useful in at least two situations. First, level-2 format could greatly reduce the size of a message by not sending the actual contents of something like a video clip. In this case, the video information is not sent until and unless the receiver wishes to view the video. The other important use of level-2 file format is to provide a "hot link" to the current version of some data. For instance, a message might contain an external-body reference to some data that is updated hourly. The receiver of the message would see valid data at the time the message is read, instead of the data as of the time the message was sent.

- Level 3. This is a MIME file. All references to external multimedia data files are replaced by encoded data. See the level-3 file representation in Fig. 4a.
- Level 4. This is a MIME file with an outgoing mail address and header information (from the mail user agent) prepended to it. At this level the data file is ready to be transported to the network services.

Vuemime accepts command line options to transform data in either direction and between any levels from 0 to 3. Level 4 is generated solely by the mail user agent (mmsend) for outgoing mail. The HP MPower component (vuepad) simply indicates to vuemime on the command line the level of the file being transformed and the level to which it is to be formatted.

Vuemime generates only base-64 encoded data when going from a level-2 (or lower-level) file to a level-3 (MIME) file even though it can accept other types of encoded data when going from a level-3 file to a lower level. Also, vuemime only generates flat multipart and mixed level-2 and level-3 messages even though it can accept and digest (via metamail) nested level-3 messages generated by a non-HP MPower MIME system.

Mimetypes. Transformation to and from raw data by vuemime is controlled by a file called mimetypes. The mimetypes file is a cross between metamail's mailcap files and HP VUE's file-type definition files. Table I shows the contents of the mimetypes file supplied with HP MPower 1.0.

Vuemime uses the data in the first column to map level-0 (raw data) files to the content type for files that are being transformed to a higher level. The third column is the content-type value inserted in files of level 1 and above. For level-1 files and above that are being transformed to level-0 files, the column 2 entry associated with a particular content type is used to determine the file name extension to be applied to the new raw data file.

Conclusion

Our multimedia email project is an example of a highly leveraged effort that combined existing blocks of functionality, some modifications, and new "glue" to meet a market need quickly.

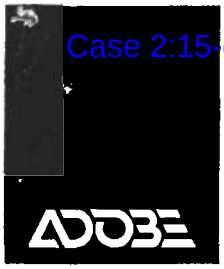
Table I
Contents of the HP MPower Mimetypes File

Search Pattern for Raw Data Files	File Extension for New Raw Data Files	Content-Type Value
.*\.xwd	%s.xwd	image/x-xwd
.*\.xd	%s.xd	image/x-xwd
.*\.tif	%s.tif	image/x-tiff
.*\.tiff	%s.tiff	image/x-tiff
.*\.eps	%s.eps	image/x-eps
.*\.pcl	%s.pcl	image/x-pcl
.*\.gif	%s.gif	image/gif
.*\.jpg	%s.jpg	image/jpeg
.*\.jpeg	%s.jpeg	image/jpeg
.*\.pm	%s.pm	image/x-xpm
.*\.xpm	%s.xpm	image/x-xpm
.*\.bm	%s.bm	image/x-xbitmap
.*\.xbm	%s.xbm	image/x-xbitmap
.*\.bmf	%s.bmf	image/x-bmf
.*\.ps	%s.ps	application/PostScript
.*\.txt	%s.txt	text/plain
.*\.au	%s.au	audio/basic
.*\.l16	%s.l16	audio/x-Linear16
.*\.l8	%s.l8	audio/x-Linear8
.*\.lo8	%s.lo8	audio/x-Linear-8Offset
.*\.wav	%s.wav	audio/x-microsoft-RIFF
.*\.snd	%s.snd	audio/x-NeXT
.*\.u	%s.u	audio/x-MuLaw
.*\.al	%s.al	audio/x-ALaw
.*\.Z	%s.Z	application/x-compress
.*\.tar	%s.tar	application/x-tar
.*\.unk	%s.unk	application/octet-stream

Acknowledgments

The multimedia mail team offer a special thanks to Gabe Beged-Dov, a former HP engineer, for discovering the leverage possibilities in the MIME internet standardization efforts.

EXHIBIT B



Adobe Systems
Incorporated

THE FUTURE OF POSTSCRIPT LANGUAGE TECHNOLOGY FROM ADOBE SYSTEMS

New applications and an expanded role for Adobe's industry-standard technology for communicating visual information

September 1991

POSTSCRIPT®
Software From Adobe

Since its introduction in 1985, technology developed at Adobe has had a profound impact on computer printing and publishing. It has solved crucial problems for software developers and hardware manufacturers, has brought 500 years of printing and publishing techniques to computer users, and has given users the power to communicate visual information more clearly and elegantly than ever before possible.

Adobe envisions new applications and an expanded role for its PostScript technology in the 1990s. The company plans to continue introducing technology that solves fundamental computing problems; makes computers more powerful communication devices; and removes obstacles preventing users of different types of computers, operating systems, applications and output devices from exchanging, displaying and printing files easily.

The future direction of PostScript technology and the problems it can help solve are closely related to the problems it solved when it first appeared.

HOW POSTSCRIPT® SOFTWARE CHANGED THE PRINTING AND PUBLISHING INDUSTRY

The key elements of Adobe's technology are the PostScript language and the PostScript interpreter, a software program that interprets page descriptions in the PostScript language. When they were introduced in 1985, there were no effective page-description standards, popular typefaces were used only with specific typesetting equipment, producing high-quality visual materials was restricted to specialists, and the cost of producing most high-quality documents was prohibitive.

In its brief history, the PostScript language and the many corporations and individuals working with PostScript products have changed all that. The PostScript language has become the industry-standard page description language, the major type libraries are available in the Adobe Type 1 format, and the cost of producing high-quality printed material has dropped substantially. As a result, the PostScript language is becoming a part of the basic fabric of the printing, publishing and computing industries.

PostScript devices helped change the electronic printing and publishing landscape because they were the first desktop printers that could print high-quality text and graphics on the same page; and the first that could quickly build and print smooth, sharp type at any size—something most experts thought was simply impossible. As devices that enabled users to accomplish both those publishing tasks, PostScript printers became one of the indispensable enabling technologies behind the desktop publishing revolution.

Acceptance of the PostScript language as a standard also benefitted software developers and hardware manufacturers. Instead of having to write dozens of printer drivers to support dozens of printing devices with different protocols, developers could suddenly write a single driver that supported every PostScript device. Hardware manufacturers, in turn, could introduce products incorporating PostScript software—the most advanced imaging technology available—and be guaranteed the devices would be compatible with thousands of applications that support PostScript devices.

Adobe technology also helped establish the clear advantages of open computing, an approach to designing and integrating systems that gives users the freedom to combine the hardware and application software they need to perform a specific task, with the assurance that they will be compatible. The PostScript language and PostScript

software also give users the freedom to print a document on any PostScript output device—regardless of the computer or PostScript-supporting application used to create it—without having to re-format it. That capability of creating "device-independent" documents lets users create a document once, on the system best suited to the task, then output the document on any output device that meets their needs for throughput, resolution, finishing features and quality.

POSTSCRIPT TECHNOLOGY TODAY

PostScript software technology has become an industry standard for electronic printing and publishing because PostScript output devices are extremely cost-effective tools for designing and producing printed materials. The standard was established in partnership with major computer equipment and peripheral vendors, which are shipping more than 200 PostScript output devices, and with leading application software companies, which have included support for PostScript devices in more than 5,000 products. More computer products support the PostScript language than support any other electronic imaging model.

Adobe has continued to revise and extend the PostScript language since its introduction in 1985. In mid-1990, the company announced PostScript Level 2, the next generation of the language. PostScript Level 2 delivers improved performance, adds new features and functionality, and is easier for software developers to use. It also incorporates previously announced extensions to the language for color, composite fonts (large character sets) and the Display PostScript® system.

While the principal application for the PostScript language to date has been as a page description language, it is in fact a general computer language with far wider application. Because the language is *device independent, platform independent, operating system independent* and *resolution independent*, it is ideally suited to serve as a **universal language for communication** among computer systems and output devices as well as for displaying and printing visual information on the widest possible range of devices.

Adobe believes the future of the PostScript language lies in expanding its role as an industry-standard printing and display language and as a universal language for computer communication. The company envisions the language developing in several ways in the 1990s, with it serving as:

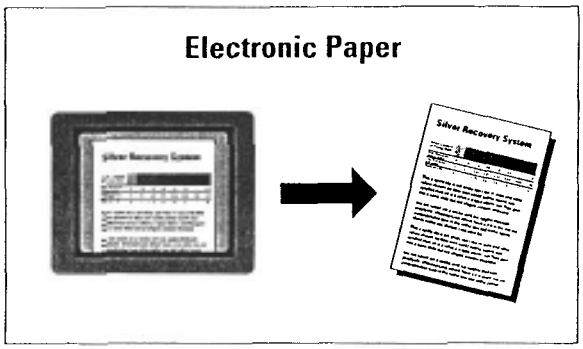
- A form of **Electronic Paper**
- An editable document interchange format
- A multimedia carrier
- An interactive document format

THE ELECTRONIC PAPER OF THE FUTURE

Predictions of a "paperless office" have never come true because there is currently no viable electronic alternative to paper. Such an alternative needs to be easier to distribute than paper and at least as easy to view and reproduce.

To make Electronic Paper feasible, the computer industry needs a device-independent format for a self-contained document that can be transmitted via network from computer to computer and location to location. Also necessary is a software program, or interpreter, that can read such a format and display or print the document on any device. Like a paper document, Electronic Paper documents will exist only in a "final form," not revisable or editable by people who receive them.

Adobe is developing both the format and the interpreter for Electronic Paper. PostScript language files today are already very close to having the features of the Electronic Paper



The goal of Adobe's Electronic Paper is to enable users to view documents on any device containing a PostScript reader and to print them on any raster output device.

format, but do not incorporate all the information, such as font outlines, needed to view and print them. Adobe believes it is the responsibility of the printer driver program to insert font information into a document, thereby taking a major step toward making a document self-contained. Future versions of printer drivers Adobe is developing will be capable of doing just that. Once a document becomes self-contained by incorporating all the underlying information describing its appearance, users will be able to view it on any device containing a PostScript "reader," or interpreter, and print it on any printer.

The PostScript language interpreter, which has already shipped on over a million devices, has the capability of being a reader of PostScript language files. Once the interpreter is slightly modified to be a reader, and development on the format is completed, Adobe plans to introduce them together as a turnkey, device-independent solution for viewing and printing formatted documents.

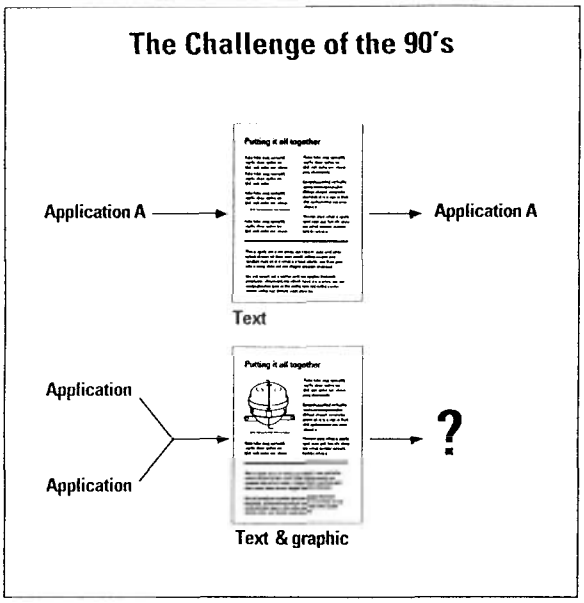
Another technology currently under development at Adobe that will help make Electronic Paper a reality is a chip that will be able to rasterize type, or create bitmapped characters from outlines, at more than 1,000 characters per second. (Current technologies rasterize at about 125 characters per second.)

The ability to create text at that speed will enable viewers to see a page in its original physical size and orientation, to scroll around the page, and to zoom in on sections of the document—all with a natural, fluid motion. The document will be rendered as if the display were a window on the page. The effect is like that of a newspaper viewed through a square window cut out of a piece of cardboard and placed over the paper. Moving the paper underneath the cardboard enables the viewer to see different areas of the paper through the window. PostScript software, by constructing an ideal representation of a page (that is, a resolution-independent representation) then modifying it for viewing or printing on a specific device, is capable of enabling users to view documents in precisely this way—a way that turns a PostScript file into an Electronic Paper document.

POSTSCRIPT LANGUAGE FILES AS AN EDITABLE DOCUMENT INTERCHANGE FORMAT—INTERCHANGE POSTSCRIPT (IPS)

A long-standing problem for users is the difficulty of combining several types of documents from several sources into one compound document. Imagine, for instance, that you're putting together a marketing plan for a new product and you need to combine several pieces of information: text describing the plan, a spreadsheet table showing projected revenues at various price points, a map illustrating projected sales by region, and photos of packaging.

All those pieces exist as computer files, but were created in different applications running on different computing platforms controlled by different operating systems. No technology currently available can serve as a common *interchange format* that will allow you to



Adobe is preparing a solution to a long-standing problem — How to enable users to combine and edit documents from several applications, regardless of the application, operating system or platform used to create the document.

import all those files, complete with formatting, into an application on your computer so you can edit them and create a single compound document.

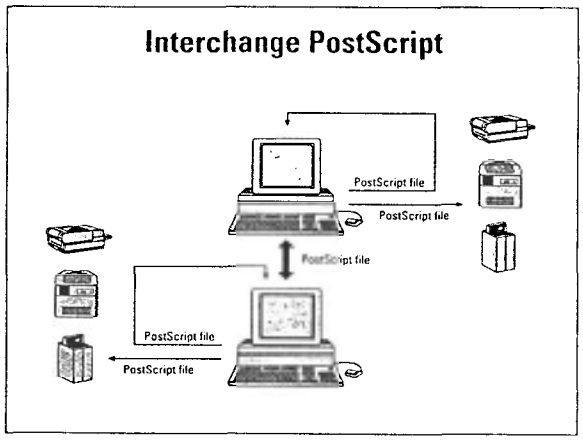
The need for this capability—the capability to exchange editable documents—is growing because today's typical corporate computing environment is made up of a variety of systems from a variety of manufacturers, and documents created on one system in one application are often impossible to view, edit and print on another system.

Current technology does permit users to import formatted files into applications other than the one used to create them, but such files are rarely editable. For today's users to exchange formatted, editable documents, they must have the same application, the same version of that application, and the same operating system. They often do not, especially if they work for different organizations.

Adobe's concept of Electronic Paper, described in the preceding section, is an intermediate stage along the way towards the ultimate goal—an editable document interchange format that contains both the final form and an editable form of a document.

Document-centric vs. operating-system-centric approaches. Several companies, including Microsoft, GO, and Patriot Partners are trying to solve the problem of exchanging formatted, editable documents by developing next-generation operating systems that handle a wide variety of applications. The drawback to such an approach is that it requires all users who want to exchange editable information to have the same operating system. The continuing proliferation of operating systems and computing platforms makes that level of uniformity unlikely.

Adobe believes the answer lies in creating self-sufficient documents that users can exchange, view, edit and print, regardless of the operating system or applications residing



Documents in the Interchange PostScript format can be viewed and edited on any platform with a PostScript interpreter, exchanged among platforms, and printed on any PostScript output device.

on their computer. This *document-centric* approach contrasts with the *operating-system-centric* approach being pursued by Microsoft and others.

In the document-centric approach, the document—not the application or the operating system—contains all the information needed to perform operations on itself. The document becomes the *artifact* of the application and is capable of being exchanged among applications and platforms.

In order for documents to serve as a self-sufficient medium of exchange, they must contain all necessary underlying semantic information. A spreadsheet, for example, contains numbers in columns and rows as well as formulas—not visible in the printed version of the spreadsheet—that generate those numbers. Similarly, an Adobe Illustrator® 3.0 document consists of objects composed of curves, lines, colors, groups and other elements. The document generates those objects as printed output and contains underlying, device-independent information describing them. An Illustrator document thus comprises not only what appears in its printed form, but also all the semantic information that generates that output.

The capability to create documents that include a final form and all the information necessary to edit the document gives the PostScript language, as evidenced in Illustrator 3.0, the potential to serve as the enabling technology for a document-centric solution.

Interchange PostScript. Adobe is currently working on extending the concepts in Illustrator 3.0 to make them the foundation for a universal editable file format for entirely self-sufficient documents.

The name for the format is *Interchange PostScript,* or IPS. IPS will be a file format that can be created by any application program and understood by any other program. Adobe's plan is to write a specification for IPS which it will make available free to application software vendors so that they may incorporate support for it into their products.

To return to the example, you could write your marketing plan easily if all the applications whose files you need to import supported the IPS format. The people creating those files would simply save them in the IPS format, and you could bring them into your application with all their formatting intact. You could then edit them as necessary and combine them into a single document.

Approaches for Exchanging Formatted, Editable Documents

Approach	Advantages	Disadvantages
Operating-system centric (O/S)	No need for O/S-independent file format	Everyone needs same O/S and same version of O/S
	No need for O/S-independent file format reader	Same O/S must run on all platforms, PC, Mac, UNIX, mainframes
Document-centric	Documents can be viewed on any platform with a file format reader and printed on any device	Application developers must support the format as an option for saving files
		Adobe and its partners must solve technical problems related to device-independent viewing and editing of documents containing text, graphics and photographs

This table briefly summarizes the advantages and disadvantages of the two approaches.

The PostScript language is the logical choice to become the technology behind such an interchange format because it can describe any kind of text, line art or scanned image that can be stored in a computer. In addition, it's already an industry-wide standard in electronic printing and publishing.

In devising a solution to the problem of displaying, as well as printing, documents in a device-independent manner, Adobe is solving the other half of the problem it solved in 1985, when PostScript software began enabling users to print documents on any PostScript device. Once the display solution is in place, users will truly be able to print and view anywhere with PostScript technology.

THE POSTSCRIPT LANGUAGE AS A CARRIER OF MULTI-MEDIA INFORMATION

Multi-media computing is a developing technology that promises to increase the power of the computer to educate and communicate. In multi-media computing, programs combine text, images, color, sound and full-motion video to communicate more effectively than would be possible without such a combination of tools.

The PostScript language is currently structured to be able to carry information about text, graphical shapes and sampled images. But its structure can be extended to carry other types of visual and audio information. Extended in those ways, it can become part of the foundation for multi-media computing and can help that technology realize its potential.

The definition of a document in the document-centric approach applies not only to paper documents and electronic files, but also to other self-contained carriers of visual information, such as a commercial television newscast. A "document" in the PostScript language is therefore an electronic file containing any or all of the following elements: text, line art, scanned or sampled images (photographs), sound and full-motion video. A newscast thus qualifies as a document being exchanged between different systems. Because they can be transformed into a series of digital images, video programs such as newscasts can be carried and enhanced by the PostScript language.

Using that capability, PostScript technology could facilitate advances in mass communication, such as a newspaper distribution system where newspaper companies send subscribers electronic versions of their paper to be read on a television screen. An existing video editing system called Video F/X enables a PostScript file to be divided into multiple pages, where each page represents a video frame. Building on that system, Adobe could use PostScript software to make the text on a television screen smoother and more readable, thereby making even newspaper-size print easy to read. Such an improvement makes it more feasible for newspapers to be delivered and read in electronic form.

The PostScript language can already carry audio information, as demonstrated by applications for NeXT computers, which allow users to add voice messages to documents. Many other companies are working on multi-media applications that will produce new kinds of complex files. Adobe's goal is not to build a standard product, but to establish a methodology for document interchange.

POSTSCRIPT LANGUAGE FILES AS INTERACTIVE DOCUMENTS

An interactive document is one that contains information instructing the user how to interact with the document's contents. It can guide you in finding information it contains and in customizing itself to suit your purposes. For example, an interactive document can guide the user in creating a training

presentation that incorporates several types of files, including text, graphics, line art and even full-motion video. Simpler interactive documents that link related information and allow users to define the relationships between pieces of information include address books, sales organizers and product documentation.

One existing format for such documents is the HyperCard language from Apple. A drawback to HyperCard documents is that are device-dependent, working only on the Apple Macintosh.

The PostScript language is more powerful than the HyperCard language and is capable of carrying information on making a document interactive. It is possible to build interactive documents using the PostScript language that can be viewed and printed on any device that contains PostScript software.

Adobe is currently exploring means to take advantage of the potential of the PostScript language as a format for interactive documents.

ADOBE'S STRATEGY

Adobe's approach to extending the PostScript language in the 90s will be to develop products that take advantage of the capabilities and potential inherent in the language. The company will also continue to enhance the enabling technology that underlies Adobe products, such as device-independent color in PostScript Level 2 and the file interchange capabilities of its Encapsulated PostScript file format. Adobe will also continue to work with OEMs, ISVs and users to find the best solutions for the industry as a whole.

The PostScript language has already become an industry standard and a proven solution to the long-standing problem of integrating high-quality text and images in computer-generated documents. Extending the functionality of the language into new areas will require substantial research and development, but the payoff can also be substantial. Adobe is prepared to undertake that research, since there are no limitations inherent in the language to prevent it from serving as the foundation for Electronic Paper, an editable interchange format, multi-media carrier and interactive document. If Adobe succeeds, the whole industry benefits, and users will find that computers have moved a giant step closer to becoming the communication tools they want them to be.

Adobe Systems
Incorporated
1585 Charleston Road
P.O. Box 7900
Mountain View
California 94039-7900

EXHIBIT C

United States Patent [19]

Isle et al.

[11] **Patent Number: 4,931,950**

[45] **Date of Patent: Jun. 5, 1990**

[54] **MULTIMEDIA INTERFACE AND METHOD FOR COMPUTER SYSTEM**

[75] Inventors: **Brian A. Isle**, Isanti; **Charles P. Bloom**, Eagan; **Arch W. Butler**, Minneapolis; **David Spoor**, Eden Prarie, David J. Wunderlin, New Hope; Renee Bedros, West St. Paul, all of Minn.

[73] Assignee: **Electric Power Research Institute**, Palo Alto, Calif.

[21] Appl. No.: **223,499**

[22] Filed: **Jul. 25, 1988**

[51] **Int. Cl.**[5] **G06F 15/46**; G06F 11/30

[52] **U.S. Cl.** **364/513**; 364/188; 364/551.01; 364/579; 364/514; 364/200; 364/274.7; 364/275.7; 381/110

[58] **Field of Search** ..... 364/513, 184–187, 364/138, 139, 188, 189, 550, 551.01, 579, 580, 513.5, 200 MS File, 900 MS File, 514, 518, 521; 381/41, 42, 43, 110; 371/29.1

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,578,555	3/1986	Inoue	364/185 X
4,602,134	7/1986	Atkinson et al.	364/189 X
4,617,661	10/1986	Futaki et al.	371/29 X
4,644,478	2/1987	Stephens et al.	364/188 X
4,644,479	2/1987	Kemper et al.	364/554 X
4,649,515	3/1987	Thompson et al.	364/900
4,654,852	3/1987	Bentley et al.	371/29
4,670,848	6/1987	Schramm	364/900 X
4,757,463	7/1988	Ballou et al.	364/551.01 X
4,776,016	10/1988	Hansen	381/42
4,816,208	3/1989	Woods et al.	364/188 X

OTHER PUBLICATIONS

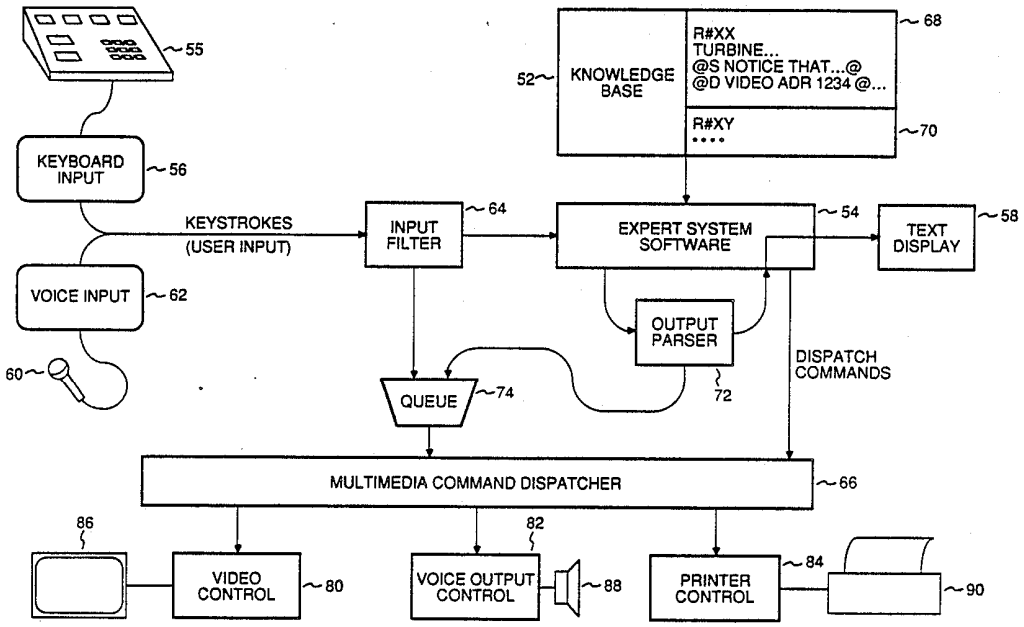
"Gas Turbine Expert System"; EPRI Identification No. AP86-11; Video Tape; 1986, Dec.; Electric Power Research Institute; Palo Alto, Calif.

Primary Examiner—Joseph Ruggiero
Attorney, Agent, or Firm—Flehr, Hohbach, Test, Albritton & Herbert

[57] **ABSTRACT**

A multimedia interface presents information and receives user commands for a computer system. The multimedia interface operates in parallel with another application software module, such as an expert system. To add multimedia features to the application software module, the module is modified so as to generate multimedia commands at the same time as it displays text on a text monitor. The multimedia commands, which are held in a queue, provide additional information in the form of video images and generated speech corresponding to the displayed text. In addition, the multimedia commands are split into at least two sets: one set which is dispatched to the user substantially immediately after displaying the corresponding text, and one set which is dispatched only upon request by the user. In the preferred embodiment, the multimedia interface presents information to the user through text, graphics, video speech production, and printed output. User inputs are made through a special-function keypad and voice recognition. The preferred embodiment is a portable expert system which fits in a single portable suitcase sized package.

35 Claims, 8 Drawing Sheets



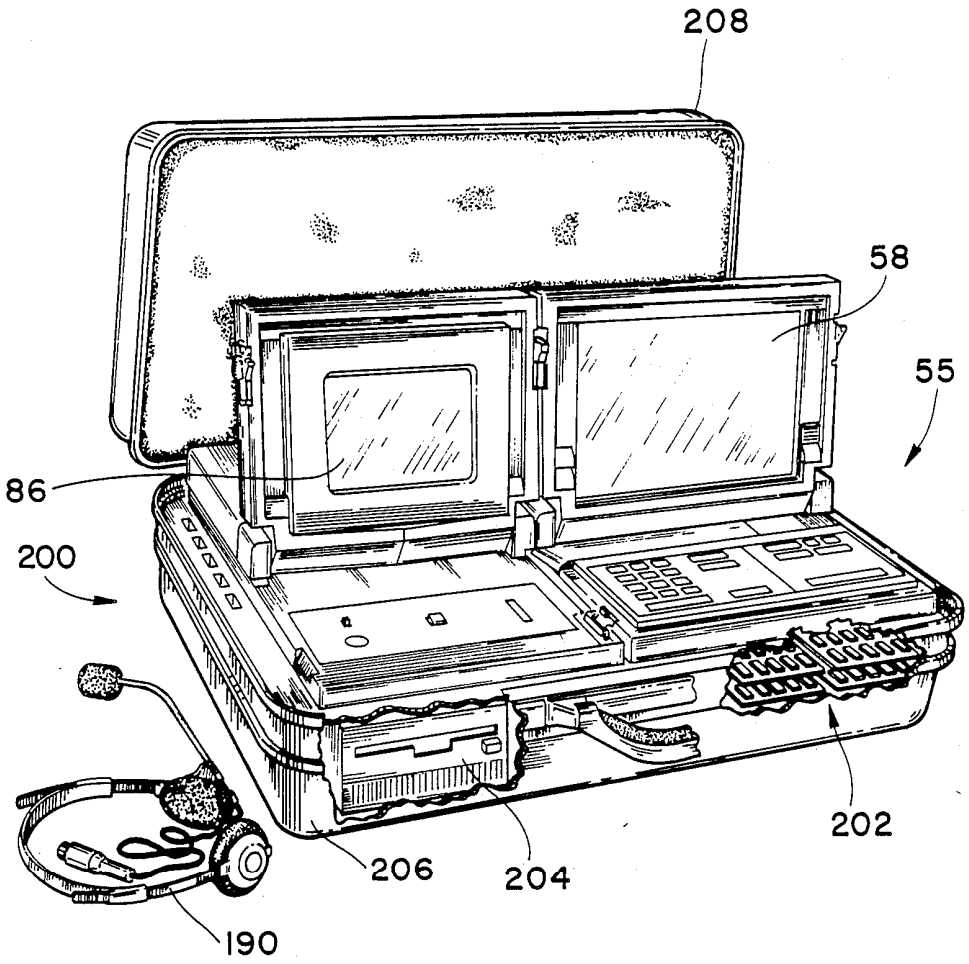


FIG. 1.

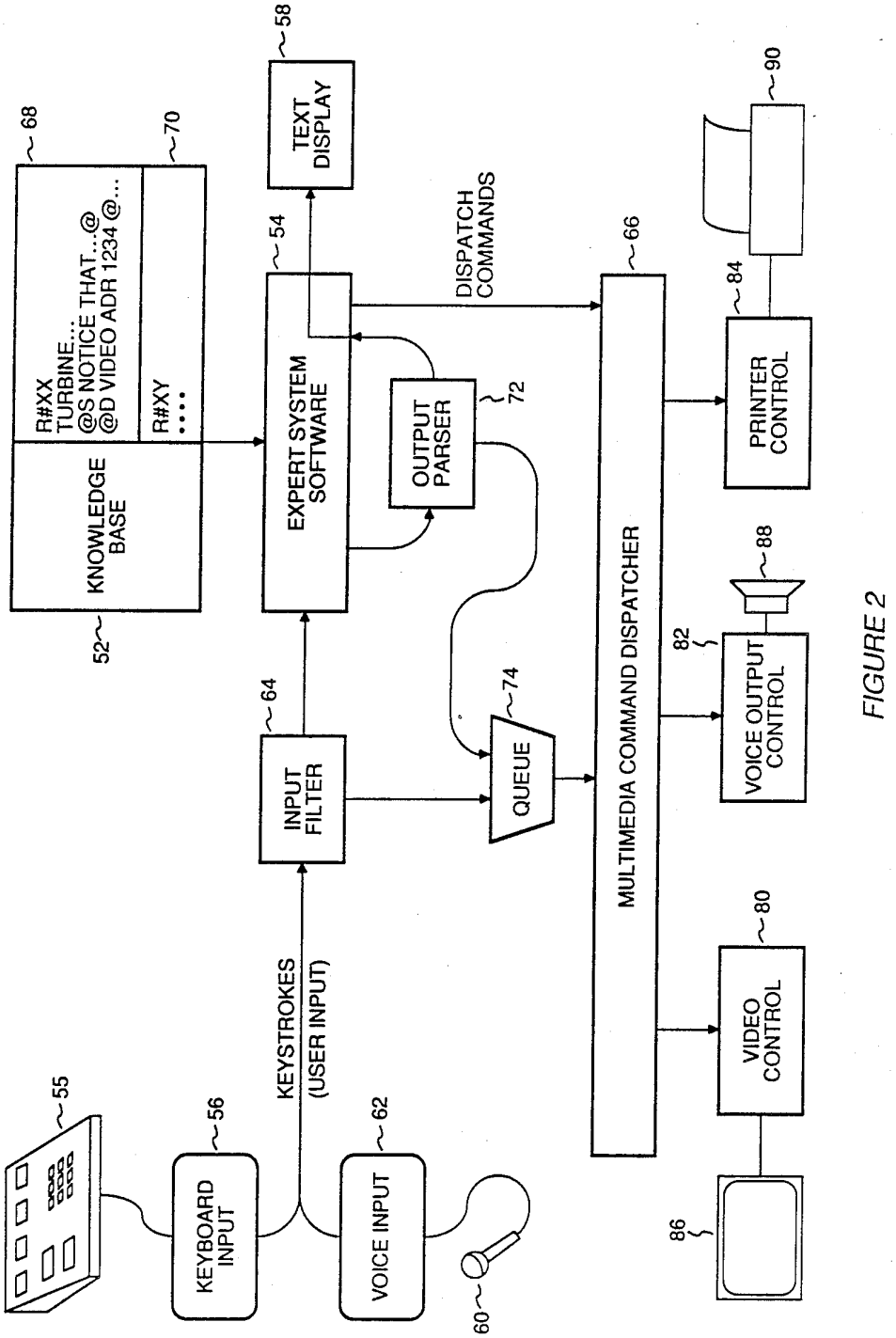

58
TEXT DISPLAY
68
70
R#XX TURBINE...
@S NOTICE THAT...@
@D VIDEO ADR 1234 @...
R#XY
....
KNOWLEDGE BASE
52
54
EXPERT SYSTEM SOFTWARE
DISPATCH COMMANDS
66
90
84
PRINTER CONTROL
OUTPUT PARSER
72
88
82
VOICE OUTPUT CONTROL
64
INPUT FILTER
74
QUEUE
MULTIMEDIA COMMAND DISPATCHER
80
VIDEO CONTROL
55
56
KEYSTROKES (USER INPUT)
62
KEYBOARD INPUT
VOICE INPUT
60
86

FIGURE 2

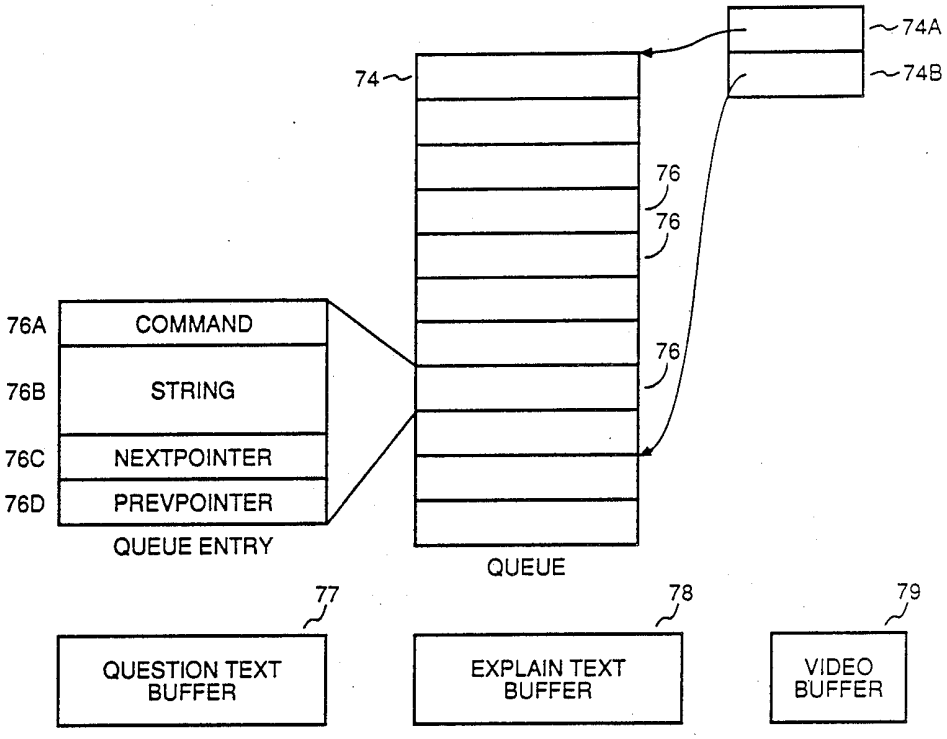

74A
74B
74
76
76
76
76A
COMMAND
76B
STRING
76C
NEXTPOINTER
76D
PREVPOINTER
QUEUE ENTRY
QUEUE
77
78
79
QUESTION TEXT BUFFER
EXPLAIN TEXT BUFFER
VIDEO BUFFER

FIGURE 3

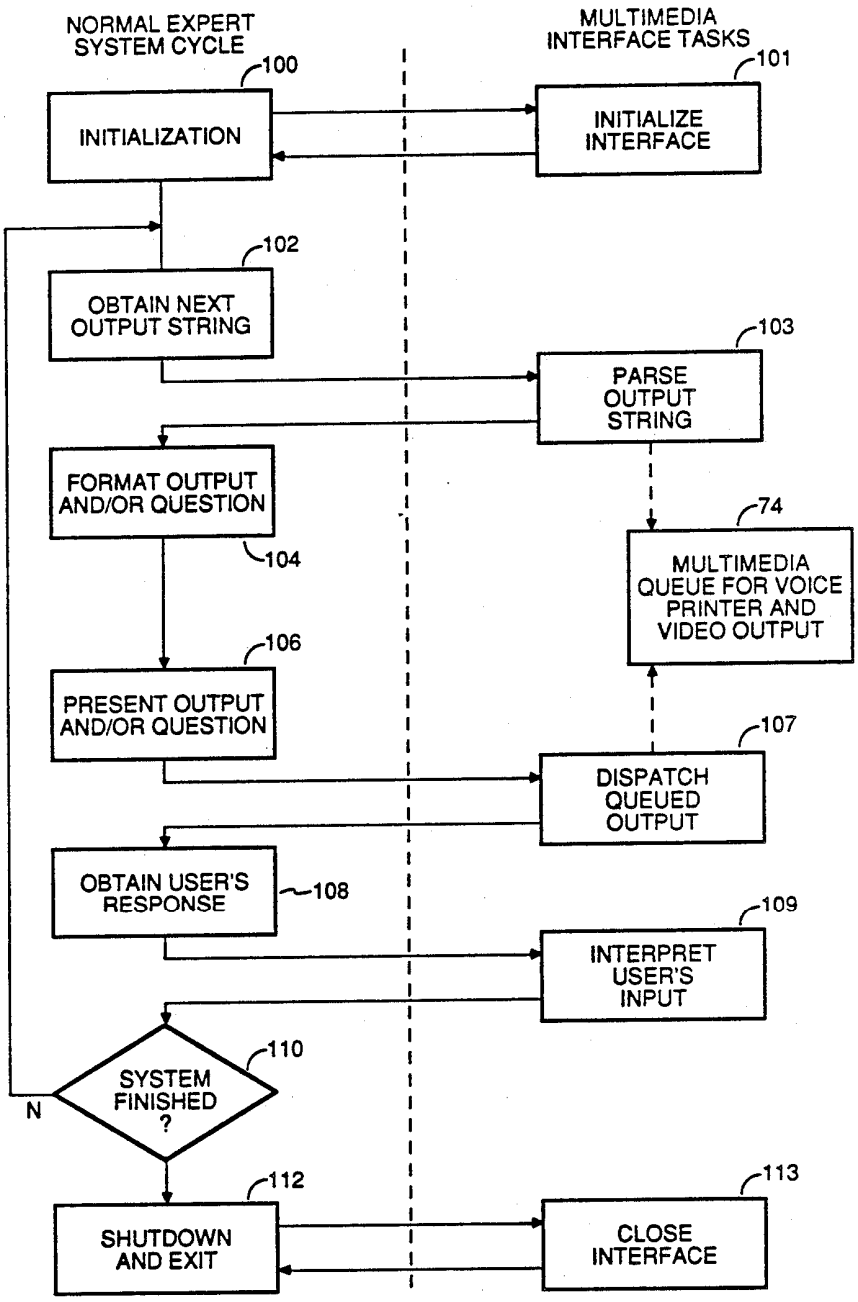

NORMAL EXPERT SYSTEM CYCLE
MULTIMEDIA INTERFACE TASKS
100
INITIALIZATION
101
INITIALIZE INTERFACE
102
OBTAIN NEXT OUTPUT STRING
103
PARSE OUTPUT STRING
FORMAT OUTPUT AND/OR QUESTION
104
74
MULTIMEDIA QUEUE FOR VOICE PRINTER AND VIDEO OUTPUT
106
PRESENT OUTPUT AND/OR QUESTION
107
DISPATCH QUEUED OUTPUT
OBTAIN USER'S RESPONSE
108
109
INTERPRET USER'S INPUT
110
SYSTEM FINISHED ?
N
112
SHUTDOWN AND EXIT
113
CLOSE INTERFACE

FIGURE 4

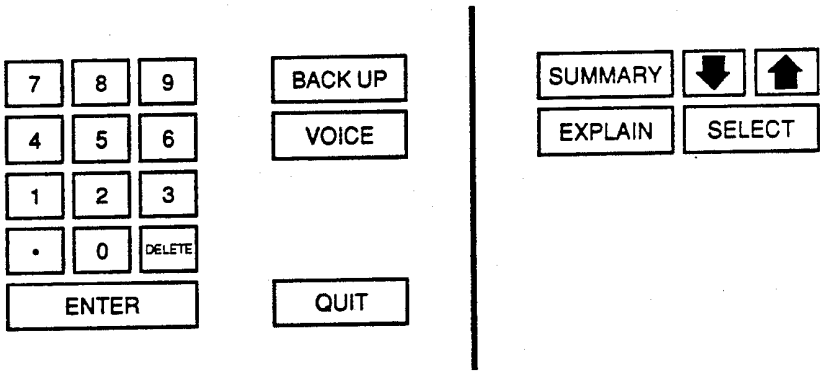


FIGURE 5A

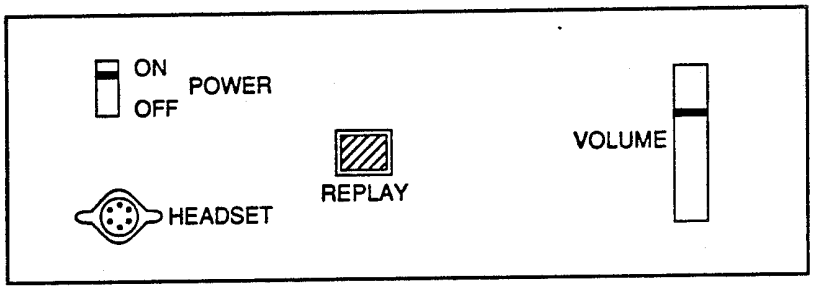


FIGURE 5B

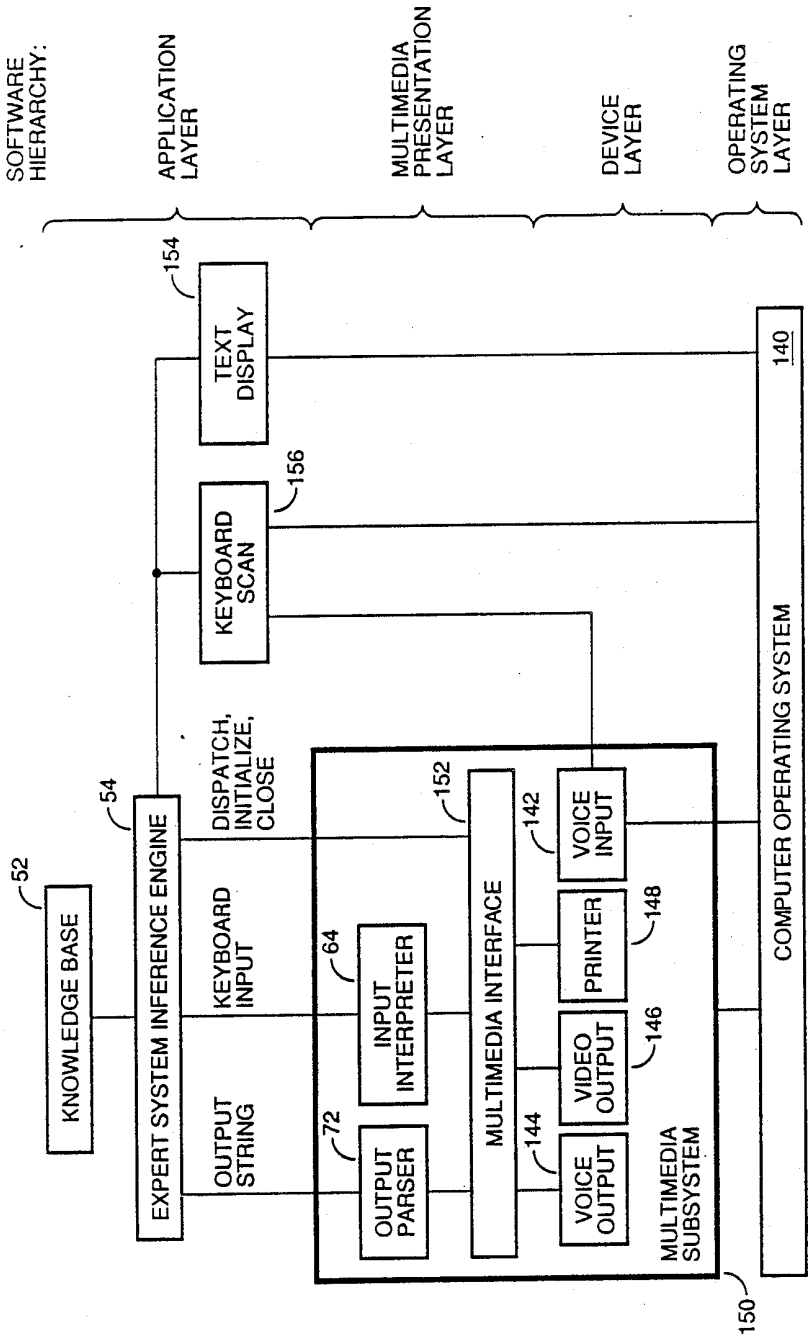

SOFTWARE HIERARCHY:
APPLICATION LAYER
MULTIMEDIA PRESENTATION LAYER
DEVICE LAYER
OPERATING SYSTEM LAYER
154
TEXT DISPLAY
156
KEYBOARD SCAN
140
COMPUTER OPERATING SYSTEM
DISPATCH, INITIALIZE, CLOSE
54
52
KNOWLEDGE BASE
EXPERT SYSTEM INFERENCE ENGINE
KEYBOARD INPUT
OUTPUT STRING
64
INPUT INTERPRETER
72
OUTPUT PARSER
152
MULTIMEDIA INTERFACE
142
VOICE INPUT
148
PRINTER
146
VIDEO OUTPUT
144
VOICE OUTPUT
MULTIMEDIA SUBSYSTEM
150

FIGURE 6

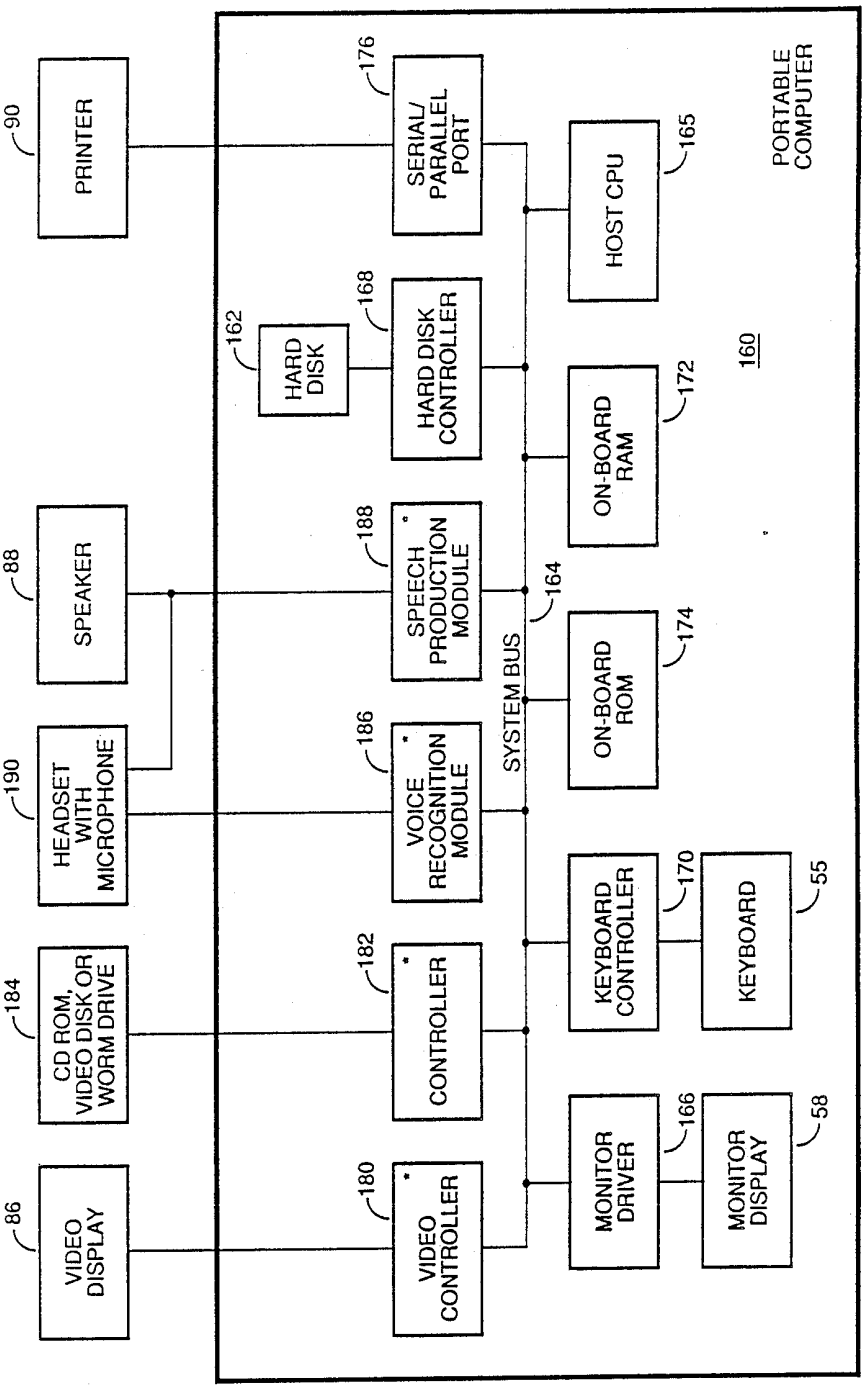

PRINTER
90
SERIAL/ PARALLEL PORT
176
HOST CPU
165
PORTABLE COMPUTER
160
HARD DISK
162
HARD DISK CONTROLLER
168
ON-BOARD RAM
172
SPEAKER
88
SPEECH PRODUCTION MODULE
188
SYSTEM BUS
164
ON-BOARD ROM
174
HEADSET WITH MICROPHONE
190
VOICE RECOGNITION MODULE
186
KEYBOARD CONTROLLER
170
KEYBOARD
55
CD ROM, VIDEO DISK OR WORM DRIVE
184
CONTROLLER
182
MONITOR DRIVER
166
MONITOR DISPLAY
58
VIDEO DISPLAY
86
VIDEO CONTROLLER
180

FIGURE 7

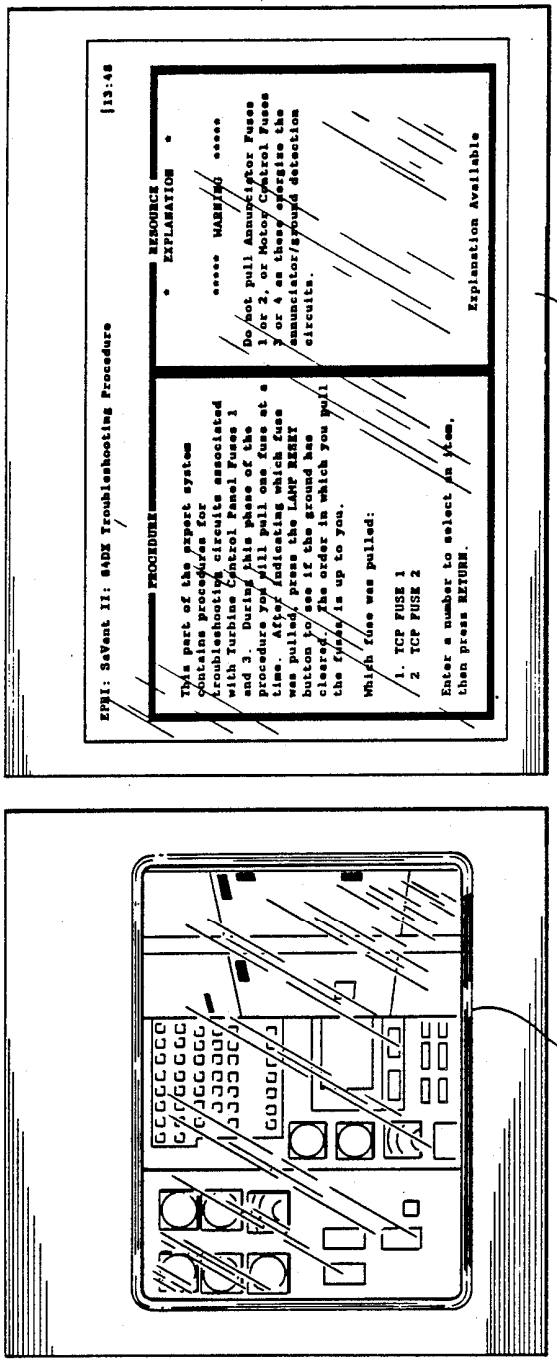

EPRI: SaVant II: SADE Troubleshooting Procedure
13:48
PROCEDURE
This part of the expert system contains procedures for troubleshooting circuits associated with Turbine Control Panel Fuses 1 and 3. During this phase of the procedure you will pull one fuse at a time. After indicating which fuse was pulled, press the LAMP RESET button to see if the ground has cleared. The order in which you pull the fuses is up to you.
Which fuse was pulled:
1. TCP FUSE 1
2 TCP FUSE 2
Enter a number to select an item, then press RETURN.
RESOURCE
EXPLANATION
***** WARNING *****
Do not pull Annunciator Fuses 1 or 2, or Motor Control Fuses 3 or 4 as these operate the annunciator/ground detection circuits.
Explanation Available
58
86

FIG. 8.

MULTIMEDIA INTERFACE AND METHOD FOR COMPUTER SYSTEM

The present invention relates to the human interface aspects of computer systems, and particularly to systems and methods for adding video and voice capabilities to expert computer systems.

BACKGROUND OF THE INVENTION

The effectiveness of many computer systems is a function not only of the application software program which interacts with the user, but also a function of how well the computer system communicates with its human users. In terms of expert systems, effectiveness is a function not only of the knowledge applied to its task domain, but also the delivery of that knowledge to the end user. In spite of the recent strides that have been made in expert system theory and development, no one has produced a truly effective delivery vehicle for communicating with the users of these systems.

Most expert systems are either implemented on a main frame computer and accessed through a terminal, or they are ported to smaller personal computers. In either case the human-computer dialogue passes through a standard text screen or monitor in one direction, and through a standard "QWERTY" keyboard in the other direction. This keyboard/monitor interface is not an interface that optimizes system performance.

In a large number of situations, the standard keyboard/monitor mode of communication is inadequate or not suitable. For instance, in situations where the user's hands are busy, keyboard input is clearly difficult. When the user must visually concentrate on a task, it may be dangerous for the user to avert his eyes so as to look at a computer screen. In other situations, computer monitors are inadequate visual media for conveying information needed by the user. For instance, computer monitors generally cannot be used to display photographic images or moving video pictures, such as a movie sequence showing what a particular piece of equipment looks like, or how to disassemble that piece of equipment to effect a particular repair.

All of the communication modules required for a multimedia interface, such as voice recognition modules, text-to-speech voice production modules, random access video memories and video displays are commercially available. However, these multimedia communication modules have generally not been incorporated into expert systems because of the difficulty of adding multimedia features to an existing expert system, or to a system for building expert systems. That is, there was an implicit assumption that one would have to totally redesign and rebuild an expert system's knowledge base and control software in order to incorporate multimedia features.

In other word words, the primary problem has been the difficulty of integrating the multimedia modules with traditional expert systems. It is therefore an object of the present invention to provide a multimedia software interface that can be added to an expert system with minimal changes to the expert system.

Another potential problem with a multimedia expert system is that increasing the number of communication modules increases the number of ways that the system can fail. It is therefore another object of the present invention to provide a modular multimedia software interface where the failure of any input or output channel does not impact the functionality of the remaining parts of the expert system.

Still another potential problem with building a multimedia expert system is that the multimedia features would be needed or useful mostly in situations requiring that the expert system be portable. While powerful portable computers are now available, these computers generally do not include voice input, voice output, video output and a printer because it has been assumed that the inclusion of all of these features would make the resulting system nonportable.

It is therefore an object of the present invention to provide a portable multimedia expert system delivery vehicle which weighs no more than 30 pounds, and is truly portable.

SUMMARY OF THE INVENTION

In summary, the present invention provides a multimedia interface, which presents information and receives user commands, for a computer system. The multimedia interface operates in parallel with another application software module, such as an expert system. To add multimedia features to the application software module, the module is modified so as to generate multimedia commands at the same time as it displays text on a text monitor. The multimedia commands, which are held in a queue, provide additional information in the form of video images and generated speech corresponding to the displayed text. In addition, the multimedia commands are split into at least two sets: one set which is dispatched to the user substantially immediately after displaying the corresponding text, and one set which is dispatched only upon request by the user.

In the preferred embodiment, information is presented through text, graphics, video, speech production, and printed output; control inputs are made through a special-function keypad and voice recognition. The preferred embodiment is a portable expert system which fits in a single portable suitcase sized package.

BRIEF DESCRIPTION OF THE DRAWINGS

Additional objects and features of the invention will be more readily apparent from the following detailed description and appended claims when taken in conjunction with the drawings, in which:

FIG. **1** is a perspective, partially cut away view of a prototype of a portable expert system incorporating the present invention.

FIG. **2** depicts a conceptual flow chart of the flow of information in the present information.

FIG. **3** is a diagram of the queue data structure used in a multimedia interface.

FIG. **4** is a flow chart of the system cycle of an expert system showing the multimedia interface tasks added to an expert system which previously did not have such tasks.

FIGS. **5A** and **5B** depict the keyboard layout for the portable expert system in FIG. **1**.

FIG. **6** is a block diagram of the software routines used in the preferred embodiment.

FIG. **7** is a block diagram of the hardware for an expert system incorporating the present invention.

FIG. **8** shows a side by side presentation of a video image and corresponding text generated by a preferred embodiment of the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

The preferred embodiment of the present invention is a multimedia interface for an expert system. It should be understood, however, that the present invention can be used with many types of computer systems and many types of application software programs.

Overview of the Preferred Embodiment

The preferred embodiment of the present invention shown in FIG. **1** is a portable computer system incorporating a multimedia implementation of an expert system known as The Expert Adviser for Combustion Turbines, developed under a contract with the Electric Power Research Institute. This is a knowledge-based system designed to aid maintenance electricians when troubleshooting gas turbine power plant control systems. This knowledge base contains procedural knowledge for operating, maintaining, and troubleshooting electromechanical equipment, controls, and accessories.

In a prototype of the present invention the expert system was used for ground-fault detection. The ground-fault detection task is a good application for an expert system because it is characterized by a complex network of symptoms, complicated fault-isolation logic, and a considerable variance in the success rate and time-to-repair attributable to the technician's expertise. The work is performed in an environment that is cramped and has temperature extremes, excessive noise, and poor lighting. These environmental factors add to the desirability of providing the worker with a cognitive aiding device.

In order for an expert system to be usable and useful in the above described working conditions, it needs to be portable and able to communicate with the user when the user's hands are busy and also when the user's eyes need to be focused on the task at hand rather than on a computer or video display. As a result, an expert system for ground fault detection in gas turbine plants was determined to be a good test vehicle for the present invention.

Expert systems, as well as other computer systems, accept input from and present information to the end user in the form of a dialogue. A two-way exchange of information with an on-site, real-time system enhances the end user's perception of the system as an expert consultant. The system leads the user through a sequence of procedures culminating in the arrival at some desired end state, such as the answer to a question or a particular problem, an explanation of how a particular portion of a machine or system works, or an explanation of how to repair a particular piece of equipment.

The exact sequence of procedures performed and information provided by an expert system will depend on each user's needs and proficiency, but the end state reached should be user independent. From an expert user, the system may only request the minimum information needed for task completion. For less experienced users, the system can explain its requests by providing the user with additional information about the locations of various pieces of equipment and standard procedures, and can elaborate on acceptable responses.

The inventors have determined that the media options required for the most effective interaction between the user and an expert system are:

1. Input of user-commands via:
   a. fixed function keys;
   b. virtual (variable) function membrane keypads;
   c. voice recognition for receiving voiced user-commands.
2. Output of text, video images and speech via:
   a. text display;
   b. video display for user-graphics and interactive video imagery;
   c. speech generator;
   d. printer for printing hard copy of text and/or images.

The system's voice recognition vocabulary is redundant with the keyboard, the audio speaker is redundant with the text display, and the printer is redundant with and an enhancement to the text display. Redundancy does not mean "identical with." Each input/output mode is customized to the manner in which humans best utilize information on that particular channel. Any redundant or primary feature may be disabled without greatly degrading the functionality of the system.

Referring to FIG. **2**, there is shown a conceptual flow chart of an expert system **50** incorporating the present invention. The basic components of the expert system **50**, excluding the multimedia features of the present invention, are a knowledge base **52** (i.e., a database of information), an expert system software module **54**, a keyboard **55**, a keyboard input module **56**, and a text display module **58**. The keyboard **55** in the preferred embodiment includes several special function keys to facilitate ease of use and the text display module is a high quality flat screen computer monitor of the type commonly found in portable computers.

The user can enter commands manually using the keyboard **55**, and orally by speaking into a microphone **60**. The oral commands are decoded by a voice input module **62** coupled to the microphone **60**. As explained below, the set of oral commands accepted by the system is a superset of the commands which can be entered using the keyboard **55**.

Inputs from the user are routed by the expert system software **54** to an input filter **64**, which stores multimedia commands from the user in a queue **74**, and passes the remaining user inputs back to the expert system software **54**. As will be explained in more detail below, the queue **74** stores multimedia commands generated by both the user and the expert system software **54**.

The expert system software module **54**, sometimes called an inference engine, interprets the user's keyboard and voice inputs and selects a new set of information from the knowledge base **52** based on the current "location" of the user in the knowledge base and the user's inputs.

The resulting output strings generated by the expert system are processed by an output parser **72**, which stores multimedia commands in the output strings on the queue **74** and passes the remainder of the output strings back to the expert system software **54** for display on the text display **58**.

A dispatch routine **66** controls the execution of the multimedia commands stored in the queue **74**. As shown in FIG. **2**, the multimedia dispatcher **66** calls subroutines for the system's video controller **80**, voice output controller **82** and printer controller **84** when these multimedia features are used. The resulting presentations are transmitted to a video screen **86**, speaker **88**, and printer **90**.

Simplified Expert System Model.

For the purposes of this explanation, a somewhat simplified structure of the knowledge base **52** and the expert system software module **54** is assumed. In particular, referring to FIG. **2**, it is assumed that the knowledge base **52** is made up of a large number of discrete records **68**, **70** and so on, each of which contains text to be displayed by a text display module **58**. The records in the knowledge base also contain information regarding the "context" of the record, and also some "multimedia" information which will be discussed in more detail below. In addition, it is assumed that the expert system software module **54** reacts to user inputs simply by selecting a new record or set of records in the knowledge base **52**, or by staying at the currently selected record or set of records.

As will be understood by those skilled in the art, this simplified model of an expert system permits explanation of the present invention without having to consider the many variations and complexities of expert system designs. While the preferred embodiment generally follows the above described model, it should be understood that the present invention can be used with virtually any expert system as well as many types of application software programs. For example, the present invention could be used to provide multimedia features in computer assisted training programs, and database access programs. Another use contemplated by the inventors is a calibration assistance program, in which the program assists the user calibrate a particular device by providing video and audio messages as needed by the user.

Multimedia Information in Knowledge Base.

When using the present invention, the text information in the knowledge base **52** is supplemented with additional text and information for the multimedia interface. More specifically, the author of the knowledge base can embed into the text in each knowledge base record **68** a number of special multimedia commands. Three types of multimedia commands specify three types messages to be delivered by the multimedia interface: verbal messages, text to be printed, and selected video images or video sequences.

Thus each record **68** in the knowledge base **62** can contain text to be displayed by the text display module **58** as well as messages to be presented by each of the multimedia output modules. Other special multimedia commands embedded in the knowledge base's text can instruct the multimedia interface to repeat a verbal or video message, clear the video screen, and so on.

As will be explained in more detail below, the text and multimedia commands in each record are formatted so that the information for each output module can be easily identified and separated from the other information in the record. The format and function of each of the multimedia command is also explained in more detail below.

Embedded Multimedia Commands.

Multimedia commands are embedded in the records of the knowledge database simply by inserting the commands into the portion of the record that specifies the text to be displayed on the system's text monitor. The beginning and end of each command is marked by a special delimiter, such as the at sign, "@", or any other character which is generally not used for other purposes. The first non-blank character following the beginning delimiter identifies the multimedia command.

The following is an example of the text in a knowledge base record, with an embedded multimedia command:

> The circuit breakers on the right hand panel are as follows . . . . @sLooking at the circuit panel on the right side of the cabinet, trip the third breaker from the bottom to turn off power to the . . . @

The symbols @s . . . @denote the beginning and end of an "s" command, which denotes "Question Text", and the remainder of the command is a parameter string - explanatory text that will be spoken when this record is selected for output.

An example of a record which contains a command to clear the video screen is as follows:

> Repair sequence is now complete. Close . . . . @m@

The symbol @m@denotes a command to clear the video screen.

Table 1, discussed below, shows the full set of multimedia commands used in the preferred embodiment.

Output Parser.

When the expert system software module **54** selects a new record **68** or set of records, it would normally send the corresponding output string directly to a text display module **58** for formatting and displaying the text. The present invention changes that by having the expert system module **54** send the output string in the selected record or records to an output parser routine **72**.

The parser **72** identifies the multimedia commands in the output string, removes each multimedia command from the output string and stores it in the queue **74**, and passes the remaining text, if any, back to the expert system software module **54** for processing by the text display module **58**.

More specifically, in embodiments using the above described embedded commands, the parser works as follows. The expert system software **54** passes an array of output information to the output parser **72**. The output parser **72** searches for "@" command markers, and stores the string between each set of command markers at the end of the queue **74**, thereby creating a new queue entry. As a result, all of the multimedia commands in the output array are stored in the queue **74**. The command markers and the multimedia commands are also removed from the output array so that when the output parser **72** returns control to the expert system software **54**, only text to be displayed on the text display **58** remains in the array of output information.

As will be explained in more detail below, the use of an output parser **72** as shown in FIG. **2** enables a multimedia interface to be integrated with an already existing expert system with minimal changes to the structure of the expert system's software module **54**.

Queue Data Structure and Use.

Referring to FIG. **3**, the queue **74** is a linked list of variable size queue entries **76** which store the multimedia commands generated by the expert system software **54**.

Each queue entry **76** temporarily stores a multimedia command, i.e., an output command specifier **76***a* and a variable size parameter or output string **76***b*. In terms of data structures, the queue **74** is a classic doubly linked

list of variable size items. Thus each queue entry **76** has both forward and backward pointers **76***c* and **76***d* for creating forward and backward links.

In addition, the queue **74** has two pointers called the queue header **74***a* and the queue tail **74***b*, pointing to the beginning and end of the portion of the queue **74** which is currently occupied by queue entries. These pointers are updated by the input filter and dispatch software as items are added and deleted from the queue **74**.

As shown in FIG. **3**, the queue **74** is supplemented by three buffers: a Question Text Buffer **77**, an Explain Text Buffer **78** and a Video Buffer **79**. The function of these buffers will be explained shortly.

Every multimedia command that is generated by the system is added to the end of the queue **74**. This is true regardless of whether the multimedia command is generated by the expert system software or by the user input (i.e., keyboard or voice input) modules.

Multimedia Dispatch and Output Control.

It should be noted that the multimedia commands stored in the queue **74** are not immediately and automatically executed. Execution and presentation to the user are controlled by a separate dispatch routine **66**.

More particularly, all of the multimedia commands stored in the queue **74** are executed and processed whenever a "dispatch" command is executed. As will be explained in more detail below, a dispatch command causes all of the items in the queue **74** to be executed in the same order that they were added to the queue.

"Executing a multimedia command" means (1) generating the voice, video or print output denoted by the command, and/or (2) updating a corresponding one of the buffers **77–79**. In addition, the dispatch command causes all of the queue entries **76** to be removed from the queue **74** as they are executed. Thus, after executing a dispatch command the queue is emptied.

TABLE 1

Command Symbol	Function	Keybd Input	Voice Input	Driving Software
MULTIMEDIA COMMANDS				
VOICE - QUESTION MESSAGES				
s	Speak and add text to question buffer.			X
S	Speak and add text to question buffer, and close buffer. Next "s" or "S" command erases question buffer.			X
a	Speak specified text, but do not add to question buffer.			X
m	Clear question buffer.			X
cntrl-R	Repeat generation of text in question buffer.		X	X *
VOICE - EXPLAIN MESSAGES				
x	Add text to explain buffer.			X
X	Add text to explain buffer and close buffer. Next "x" and "X" will erase explain buffer.			X
z	Speak text in explain buffer.	X	X	X *
K	Clear explain buffer.			X
VIDEO IMAGES				
d	Show specified video image or video sequence and store in video buffer.			X

TABLE 1-continued

Command Symbol	Function	Keybd Input	Voice Input	Driving Software
MULTIMEDIA COMMANDS				
r	Replay last video.	X	X	X *
k	Clear video buffer.			X
PRINTED TEXT				
p	Print specified text.			X
OTHER COMMANDS				
v	Toggle voice input mode between "mike on" and "mike off".	X	X	X *
M	Clear the video screen.			X
Z	Reset all: clear question buffer, explain buffer, video buffer and video screen.			X
cntrl-L	List Choices: Verbally list the commands that the voice input system will accept.	X	X	X *

* These commands can be generated by expert system software, but should normally be generated only by user.

This method of separating the definition of what to do (i.e., storing multimedia commands) and when to do it (by executing a dispatch command) prevents the multimedia interface from slowing down the user, because it enables the user to view and respond to text presented on the text display without having to wait for the slower video and voice outputs by the multimedia interface.

Table 1 lists the full set of multimedia commands which can be stored in the queue **74** in the preferred embodiment. As shown in the table, some of the multimedia commands are generated only by the driving software (i.e., the expert system software), while others are normally generated by the user's inputs via the keyboard or voice input system. The commands which are normally generated by user inputs can be also be generated by the expert system software, but the nature of these commands makes this unlikely to be useful in most circumstances.

Consider the following examples of what happens when multimedia commands are executed.

EXAMPLE 1

The queue contains the following multimedia commands:

Command: S	
String:	"Select 1 to see a more detailed circuit diagram. Select 0 to see other related circuits."
Command: d	
String:	Show images in files: F001 to F012, in sequence.

When this queue is executed (i.e., dispatched), the text in the first command is spoken and added to the Question Text Buffer **77**. A flag in the Question Text Buffer is set noting that this buffer is to be erased the next time that an "s" or "S" command is executed. In addition, the sequence of images specified by the second command are shown on the video display and the specification for this sequence of images is stored in the Video Buffer **79**, replacing the previous contents of the Video Buffer, if any.

The actual format of the parameter string for a video command is "FILENAME, PAUSE, START, END", where FILENAME specifies the name of a file containing a video image. When a sequence of images is to be presented, the PAUSE parameter specifies the delay or pause between successive images, and START and END specify the suffixes (i.e., filename extensions) of the filenames storing the first and last images in the sequence.

For example, a sequence of video images can be stored in files with a sequence of filenames such as IMAGEX.001 to IMAGEX.024. The parameter string for the corresponding multimedia command, with a one second pause between images, would be "IMAGEX, 1, 001, 024".

EXAMPLE 2

The following multimedia commands are added to the queue immediately after the commands in Example 1 are executed:

Command: s	
String:	"Looking now at the stator of the motor, check the stator windings for burn marks or other obvious defects."
Command: K	
Command: x	
String:	"All sections of the stator must be inspected. Use a mirror . . . "
Command: d	
String:	Show image in file: F031.

When the commands in this queue are executed, the "s" command causes the text in the Question Text Buffer **77** to be deleted, the text of the "s" command to be spoken, and text of the "s" command to be added to the Question Text Buffer 77. The text previously stored in the Question Text Buffer **77** is deleted because the previous Question Text command was an "S" command, which requires that the Question Text Buffer **77** be cleared before new text is stored in that buffer. The "K" command clears the previous contents of the Explain Buffer **78** and the "x" command adds new text to the Explain Buffer **78**.

The "d" command causes the image on the video display to be replaced with a new image, and the specification for this image is stored in the Video Buffer **79**, replacing the previous contents of the Video Buffer.

The new explain text is not spoken unless and until the user requests further explanation. When the user speaks the command "Explain" into the voice input module, or presses the "Explain" key on the keyboard, a "z" command is added to the queue, and then a dispatch command is generated—which causes the "z" command to be executed and the text in the Explain Buffer **78** to be spoken.

As shown by the above examples, one of the primary uses and advantages of using the queue **74** and its associated buffers **77–79** is that it enables the expert system to set up a specified sequence of multimedia commands in advance. The dispatch command sequentially processes the items in the queue **74**, in the same order that they were added to the queue. Therefore the expert system software **54** can control the order that multimedia commands are executed and the order that information is presented to the user by controlling the order in which multimedia commands are added to the queue **74**.

The two verbal buffers, called the Question Text Buffer **77** and the Explain Text Buffer **78**, provide the flexibility needed for handling common expert system dialogue sequences. The commands listed in Table **1** enable the expert system to have complete control over the contents of the verbal Question and Explain Buffers **77** and **78**. In particular, there are commands for clearing each buffer, adding text to each buffer, and adding text to a buffer and marking that buffer so that the next time text is added to it all previously stored text is erased (see S and X commands in Table 1). Thus the contents of each verbal buffer can be added to and cleared by appropriate multimedia commands, allowing the text in each buffer to be managed through the use of a sequence of multimedia commands.

One example of the type of dialogue control provided by the dual verbal buffer scheme is as follows. The text in each multimedia Question Text command ("s", "S" and "a") is normally presented to the user, i.e., spoken or verbalized, when (i.e., just after) the corresponding text is displayed on the text monitor. The text in the Explain Text Buffer, on the other hand, is normally spoken only if the user requests additional explanation of the previously presented materials. Using this dual buffer scheme, an expert user can be spared having to listen to detailed explanations that he does not need, while making additional explanations available to less skilled users.

System Software

Referring to FIG. **4**, there is shown a basic flow chart of the operation of an expert system. The steps on the right side of the flow chart are performed by the multimedia interface, and the steps on the left side of the flow chart are those performed by the control software in a traditional expert system. Thus the operation of the prior art or traditional expert system is represented by the left side of the flow chart, skipping the items on the right side of the Figure.

It should also be noted that in the preferred embodiment all of the steps shown in FIG. **4** are performed by a single processor. In the arrangement shown, the multimedia interface tasks or routines **101**, **103**, **107**, **109** and **113** are subroutines called by the expert system software **54**.

As shown in FIG. **4**, there are only five multimedia software routines **101**, **103**, **107**, **109** and **113** that must be called by the expert system's software. In addition, the present invention makes it very easy to add the multimedia interface of the present invention to an existing expert system or expert system generator by limiting the number of places that the multimedia interface must be called.

When the expert system software performs its own initialization routine **100**, it calls the multimedia interface initialization routine **101**, which checks that all of the multimedia modules respond to standard commands and are in working order. If any of the multimedia modules are not working properly, that module is "masked out", so that all multimedia commands directed at these modules will be ignored. More particularly, the initialization routine sets each of four flag values:

VOICEIN_BOARD_GOOD
VOICEOUT_BOARD_GOOD
VIDEO_BOARD_GOOD
PRINTER_BOARD_GOOD

to a value of 1 if the corresponding module is in working order, or to a value of 0 if it is not in working order.

The multimedia modules with flags set to zero are effectively removed .from the system. During subsequent operation, the system discards (i.e., does not attempt to dispatch) multimedia commands directed at modules that have been flagged as not being in working order. When the VOICEIN_BOARD_GOOD parameter denotes that the voice input module is not working (i.e., this parameter is set to zero), no voice recognition will occur and the system will not call any of the routines which access the voice input hardware. In this way, the failure of any particular module is prevented from adversely affecting the operation of the other portions of the computer system.

After initialization, the normal processing of the expert system cycle, boxes **100–110**, begins. The first step of this cycle is an action by the expert system software: selection of a portion of the expert's knowledge base, resulting in the generation of an output string (box **102**). The output string is passed to the multimedia interface for parsing (box **103**), which removes the multimedia commands in the output string and adds them to the end of the multimedia command queue **74**.

The remaining portions of the output string are passed back to the expert system software, which formats the output string for display (box **104**) and presents (box **106**) the output string, on the system's text monitor. In addition, a dispatch command is generated (box **107**) so that all of the multimedia commands stored in the queue **74** will be executed. Typically, this will result in the generation of a verbal explanation corresponding to the displayed text. As explained above, depending on the commands in the queue **74**, a video display and printed information may also be generated.

As shown by the order of the output processing steps **103**, **104**, **106** and **107**, text information is displayed on a text monitor before the multimedia commands stored in the queue **74** are dispatched and presented to the user. This order of presentation facilitates efficient communication with the user because text monitors are generally much faster than voice and printer output, and are usually faster than video because displayed text is static while video output may show a sequence of images. From a functional point of view, however, multimedia commands are executed (i.e., presented to the user) when the corresponding text is displayed on the text display module **58**.

The output string usually includes a question to be answered by the user, providing a basis for selecting new information in the knowledge base. Thus the next step of the expert system cycle is to obtain the user's response to the previous output string (box **108**).

Inputs received from the user are initially passed to an input interpreter or parser (box **109**) so that multimedia commands from the user will be removed from the input string processed by expert system. Instead, multimedia commands input by the user are stored at the end of the command queue **74** and then dispatched immediately.

As discussed below, in the section on User Inputs, before user inputs are processed by the input interpreter, they are verbally acknowledged by speaking the name of the key which has been pressed, or by repeating the verbal command received. User inputs also generate a dispatch interrupt signal or command, which "flushes" the command queue **74** by processing any multimedia commands in the queue **74** so as to clear the command queue **74** and update the queue buffers **77–79** without physically performing the commands. The dispatch interrupt furthermore suspends output by the multimedia interface if the presentation of the previously queued outputs has not yet been completed.

The remaining portion of the user's input, after removal of any multimedia commands, is passed back to the expert system software. If the user's input indicates that the user is not yet finished using the system (box **110**), processing of the user's input is performed beginning at the top of the expert system cycle (box **102**).

However, if the user's input indicates that the user is finished using the system (box **110**), the expert system performs a shutdown and exit routine (box **112**). Part of the expert system's exit routine is a subroutine call to the Close Interface routine (box **113**) which shuts down the multimedia interface.

Adapting Expert System Software to Use Invention.

When adapting an expert system to work with the multimedia interface of the present invention, the expert system software needs to be modified in only five places.

As shown in FIG. **4**, two of the modifications to the expert system software are simply adding subroutine calls to the expert system's Initialization and Exit routines so as to call the Initialize Interface **101** and Close Interface **113** routines of the multimedia interface.

The expert system software's output routine(s) are modified in two places so that whenever the software **54** would normally display text on the text monitor, if now performs three steps: (1) it passes the output data to an output parser **72** which stores in the queue **74** the multimedia commands in the output data and returns the remaining text to the expert system software; (2) it displays the remaining text on the text monitor **58** (i.e., the normal processing of output strings); and (3) it generates a dispatch command (box **107**). To do this, subroutine calls to the Parse Output and Dispatch routines are added at appropriate places in the expert system's output software.

The last of the five modifications to the expert system software is to add a subroutine call to the Input Parser (box **109**) for processing user inputs before these inputs are processed by the inference software in the expert system.

The above described modifications to the expert system software are partially based on the assumption that multimedia commands are added to the system's knowledge base using the "embedded command" scheme described above. In particular, the placement of the subroutine calls to the output parser and the dispatch routines are based on the "embedded command" scheme.

If the multimedia commands are stored in separate data structures or files from the output text strings, then the expert system's output software would need to be modified to access the separately stored multimedia commands corresponding to the selected output text strings, and to store the selected multimedia commands in the command queue **74**. In such a system, steps of the expert system cycle in FIG. **4** could be changed as follows. At the beginning of the loop, starting at box **102**, the expert system could first perform the knowledge selection, output formatting and text output operations represented by boxes **102**, **104** and **106** before processing the multimedia commands. Then it access the separately stored multimedia commands, call a rou-

tine for storing them in the multimedia command queue **74**, and then call the dispatch routine. The resulting text and multimedia presentations to the user would look exactly the same as those generated in the preferred embodiment.

User Input and the Input Filter.

Referring to FIG. **2**, the present invention changes the handling of user inputs. An input filter or interpreter **64** receives all inputs from both the keyboard and voice input modules **56** and **62**. To minimize the impact of the multimedia interface on the expert system software **54**, all inputs are actually initially received by the same input handling routine in the expert software **54** that normally receives inputs from the keyboard. The input handling routine, however, is modified so that the first thing it does is pass each input to the input filter **64** for processing. In the preferred embodiment, the input filter **64** is considered to be part of the multimedia interface.

User inputs are parsed (inspected) by the input filter **64** so that commands specifically for the multimedia interface are processed and removed from the system's input buffer before the remaining input commands are passed to the expert system software module **54** for "normal" expert system processing. For instance, when the user enters multimedia commands to stop or repeat the display or playback of a video or oral sequence, these commands are processed by the multimedia interface, not by the expert system software module **54**.

Before a user input is processed by the input filter **64**, the user input causes a sequence of events in the multimedia interface. In particular, if the multimedia interface is in the midst of processing queued commands when the user enters an input of any type, the multimedia system flushes the queued commands (as described in more detail below), stops the outputs being generated by the multimedia output modules, and returns control to the expert system. Enabling the user to interrupt the multimedia interface is very important in a practical system because it allows the user to cut short explanations that he or she doesn't need.

Thus, if the user obtains enough information from the displayed text or the beginning portion of the spoken text to enter a new command, the present invention allows the user to interrupt the presentations by the multimedia interface simply by entering a new command through either the keyboard **55** or microphone **60**. To accomplish this function, the dispatcher **66** continually monitors for user inputs. When it sees that a new user input has been entered, it generates a "dispatch interrupt" which stops all physical performance of the queued multimedia commands, halting any video sequences currently being played and any text being spoken. It also flushes the command queue **74** by processing of the meultimedia commands in terms of updating the queue buffers **77–79** but without physicaly performing any visual or audio outputs, and removing the multimedia commands from the queue **74**.

More specifically, each user input causes the following sequence of actions to be performed. First, interrupt commands are sent to each of the output media controllers **80**, **82** and **84** to suspend output by these output media. In addition, the processing of the remaining items in the queue **74** is completed. However, when the items in the queue are processed, no output presentations are generated. Instead, each multimedia command is executed only to the extent that it affects the state of the buffers **77–79** (see FIG. **3**). Thus the resulting state of the buffers **77–79** is the same as though the output commands were processed normally, making the state of the buffers **77–79** independent of how quickly the user responds to the questions posed by the expert system software.

Next, the input filter **64** verbally acknowledges each user input by speaking the name of the key which has been pressed, or by repeating the verbal command received. In the preferred embodiment, this is done by generating a dispatch interrupt, followed by placing a simple voice text command (i.e., an "a" command) on the queue **74** with the text for the verbal acknowledgement, and then generating a dispatch command so that the queued voice command will be executed.

Finally, the user input is processed by input filter **64**. If the user input is a multimedia command, the command is placed at the end of the command queue **74** and a dispatch command is generated so that the user's multimedia command will be processed immediately. Otherwise, the user input is passed back to the expert system software **54** for further processing.

Keyboard and Voice Input.

Referring to FIGS. **5A** and **5B**, the keyboard in preferred embodiment is divided into two sections: the main section shown in FIG. **5A** is placed in front of the system's text display **58** (see FIG. **1**) while an auxiliary section of the keyboard (shown in FIG. **5B**) is located in front of the video display **86**. The keyboard section shown in FIG. **5A** is a sealed-membrane-type design with snap-dome keys made by Honeywell Inc.'s Microswitch division, impervious to water or oil, and shock resistant. The individual keys provide positive tactile feedback that can be felt through medium-weight rubber gloves. The multimedia interface provides virtually instantaneous (delay of less than **250** milliseconds) audible feedback of key actuation by verbalizing the identity of the key that was pressed (e.g., if the user presses the delete key, the voice output system says "DELETE"). Between 350 and 500 grams of force is required to actuate the keypad/keyboard switches at 75° F. The keypad can support data input of **80** keystrokes per minute.

The keyboard section shown in FIG. **5B** has a standard slide, single-pull, double-throw POWER ON/-OFF switch, a sliding knob VOLUME control, and a momentary, normally-open push-button REPLAY key. In addition, the plug for the user's headset **190** is located on the auxiliary section of the keyboard.

The keyboard is designed for single-hand operation, and no simultaneous multiple-keypress functions are required. The number of keys on the keypad have also been minimized, with the constraint that multiple keypress sequences have been eliminated. The keys are arranged on the keypad in three functional groupings, page scroll, function selection, and data entry. The layout of the numerical keys follows the telephone configuration (i.e., 1-2-3 along the top row). Labeling and graphics for the keyboard are applied with silk screening and provide a durable display. Table 2 lists the functions associated with the keys used in the preferred embodiment.

TABLE 2

KEYBOARD FUNCTIONS	
NAME OF KEYPAD	DESCRIPTION
PAGE SCROLL	
UP ARROW	SCROLL UP ONE PAGE

TABLE 2-continued

KEYBOARD FUNCTIONS	
NAME OF KEYPAD	DESCRIPTION
FUNCTIONS	
SCROLL DOWN	SCROLL DOWN ONE PAGE
SUMMARY	PROVIDE SUMMARY OF SESSION
BACKUP	GO TO PREVIOUS MENU
QUIT	EXIT FROM EXPERT SYSTEM
SELECT	SELECT DESIGNATED MENU ITEM
VOICE	TOGGLE VOICE INPUT ON/OFF
EXPLAIN	SPEAK ADDITIONAL EXPLANATORY TEXT
REPLAY	REPLAY LAST VIDEO
VOLUME	SLIDING KNOB CONTROLS VOICE OUT VOLUME
POWER	ON/OFF SWITCH FOR ENTIRE SYSTEM
DATA ENTRY	
0–9	DECIMAL VALUES
.	DECIMAL POINT
BACK UP	DELETE LAST ENTRY
ENTER	ENTER SPECIFIED VALUE OR SELECT DESIGNATED ITEM

It should be noted, however, that all of the keys (except for the POWER ON/OFF key, and the VOICE OUT VOLUME sliding knob) in the preferred embodiment are "programmable" function keys. That is, the function of the keys is easily modified through the use of software, and changing the template on the keyboard. Clearly, most systems will have data entry keys similar to those used in the preferred embodiment. Nevertheless it is useful for all or virtually all of the keys to be programmable to enable customizing the multimedia interface for use with different applications, because different applications (.e.g, different expert systems) have different commands that can be conveniently entered through the use of single keystroke.

The PAGE UP and PAGE DOWN keys scroll the text display up and down one page at a time. The data entry keys are used for entering numeric values in standard fashion.

The PREVIOUS MENU, QUIT SESSION, and SESSION SUMMARY keys are specialized functions adapted for use with The Expert Adviser for Combustion Turbines expert system. The PREVIOUS MENU key prompts the expert system to back up to the previously shown menu; the QUIT SESSION key enables the user to abort the user from the standard expert system cycle in a single step; and the SESSION SUMMARY key prompts the expert system to display a summary of the portions of the knowledge base that the user has used during the current session.

The REPLAY key generates a multimedia command that causes the system to regenerate the last defined video image or sequence of video images. The EXPLAIN key generates a multimedia command that causes the system to speak the text currently stored in the EXPLAIN TEXT buffer.

The POWER ON/OFF key turns the power for the entire computer system, including the multimedia interface, on and off. The user should press the QUIT key before turning the system's power off, so that the system can close its files and perform other functions necessary to ensure proper operation of the system. As will be understood by those skilled in the art, in an alternate embodiment the system can be modified so that pressing the POWER ON/OFF key causes the system enter a QUIT command and pause briefly before shutting off the power so as to enable a graceful exit by the expert system and multimedia software.

The VOLUME control key is a sliding knob coupled to a standard volume control potentiometer, which controls the volume of the spoken words generated by speaker **88** (see FIG. **2**) in conjunction with the voice output module **82**.

The VOICE key enables the user to toggle to status of the voice input module **62** (see FIG. **2**) on and off. When voice input is enabled or ON, the user can enter commands verbally via an input microphone. When voice input is disabled or OFF, the user cannot enter commands verbally. This is particularly useful when the user needs to converse with someone or is near someone else who is talking, and the user wants to ensure that the multimedia interface does not interpret such conversations as user commands.

Voice Recognition System Characteristics

The voice recognition system is a commercially available system (the T.I. Speech TM system made by Texas Instruments) capable of distinguishing a vocabulary of 60 words from each of the predetermined speakers. High recognition accuracy is required to prevent user dissatisfaction. In field tests, 98% of input words are correctly understood by system. In most cases, if the system fails to correctly recognize a spoken input, the system requests that the input be repeated rather than make a false recognition. The voice recognition system operates in an environment of up to 85 dB without significant loss of accuracy. A maximum of three passes are used for initial speech recognition template training and testing. The voice recognition device allows syntax node creation after training is completed. Input is via a microphone attached to a lightweight headset **190** (see FIG. **7**). Although, wireless communication between the headset and computer would be preferred, the high level of electro-magnetic static in the initial prototypes does not permit it. Instead, the headset is attached to the system by 20 feet of cable. It is anticipated, however, that wireless headsets will be used in at least some future embodiments of the invention.

Verbal input is virtually a necessity in situations where the user's hands are busy performing a manual task. Thus it is important that the user be able to verbally enter all of the commands that could be entered via the system's keyboard.

In the present invention, the verbal input vocabulary is actually larger than the set of keys on the keyboard. The addition input commands provided by the verbal input module include commands necessary for dealing with verbal input.

Table 3 lists the voice recognition vocabulary when voice input is enabled and when voice input is disabled.

As shown in Table 3, when voice input is in "mike off" mode, the voice input module responds to only two verbal commands: MIKE ON, and LIST OPTIONS. The verbal command "MIKE ON" puts the voice input module into "mike on" mode, which enables full operation of the voice input module and switches the input vocabulary used by the voice input module to the list shown in the left hand column of Table 3. Regardless of whether the voice input module is in "mike on" or **37** mike off" mode, the command "LIST OPTIONS" prompts the system to verbally list all of the verbal commands that it will accept at the current time.

TABLE 3

VOICE INPUT VOCABULARY

MIKE ON MODE	MIKE OFF MODE	EQUIVALENT KEY PAD
mike off	mike on	-none-
list choices	list choices	-none-
-none-	-none-	VOICE
zero		0
one		1
two		2
three		3
four		4
five		5
six		6
seven		7
eight		8
nine		9
ten		10
eleven		11
point		.
delete		delete
enter		enter
page up		page up
page down		page down
summary		summary
quit		quit
say menu		-none-
replay		replay
explain		explain
backup		backup

When the voice input module is in "mike on" mode, the user can "partially disable" the voice input module simply by giving the verbal command "MIKE OFF". However, "partially disabling **38** the voice input module actually only changes the vocabulary list that the voice input module will recognize. Thus, when the voice module is "mike off" mode, it does actually still listen to and interpret spoken inputs. In contrast, the VOICE key on the keyboard (shown in FIG. 5A) turns the entire voice input system on and off, thereby enabling and disabling voice input.

Software Modularity.

Referring to FIG. **6**, the software in the expert system is arranged in a set of hierarchial levels, with each of the lower levels providing services to the routines or modules at the higher hierarchial levels.

AT the lowest level, the standard operating system **140** for the computer being used provides standard features such as a disk operating system, memory management, and support routines for the computer's text display and keyboard.

At the second lowest level are the device layer modules **142–148**. The device layer modules are the control routines for the multimedia interface's input/output facilities, including control programs for voice input **142**, voice output **144**, video output **146** and printer output **148**.

The device layer modules **142–148** in the multimedia subsystem **150** are modular and independent of the other portions of the system. If any of the multimedia input or output devices is inoperative, the code for that device is shut off and does not affect the operability of the remaining portions of the expert system. More particularly, as discussed above with respect to FIG. **4**, the multimedia interface initialization routine **101** checks each of the multimedia devices to determine whether it is operative (i.e., whether it responds to a reset command). If a device is inoperative, a corresponding device availability flag is disabled, which prevents the multimedia interface from calling the corresponding device control routine.

The multimedia presentation layer shown in FIG. **6** contains all of the software which controls or directs the flow of information in and out of the multimedia subsystem **150**. As discussed above with respect to FIG. **2**, the input interpreter or filter **64** screens keyboard and voice inputs for multimedia commands, so that multimedia commands are stored in a command queue **74** while all other inputs are passed to the expert system software **54**. The output parser **72** screens output generated by the expert system software **54** so that multimedia commands in the system's outputs are stored in the queue **74** and the other outputs are sent to a text display.

The functions of the remaining portion **152** of the multimedia interface software, including the queue data structure **74**, the dispatch routine **66**, the initialization routine **101**, and the close interface routine **113** have been described above, with reference to FIGS. **2** and **4**.

Finally, the application layer in the software hierarchy of FIG. **6** contains the expert system software **54** and its knowledge base **52**. Also included in the application layer are a text display routine **154** which formats output strings for display on the system's text monitor **58**, and a keyboard scan control routine **156** which interprets keyboard inputs by the user.

In other embodiments of the invention the application layer would include whatever application software that the multimedia interface is being used with.

System Hardware

Referring to FIG. **7**, the preferred embodiment comprises a modified GRID model 1525 "laptop" or portable computer **160**. This computer has an 8-Mhz 80286 microprocessor or CPU **165** and an automatically parking 30-Mbyte internal hard disk **162**. Future models of the invention are expected to use even more powerful microprocessors and larger internal hard disks.

As is standard in microcomputer systems, the computer's system bus **164** couples all of the system's peripheral devices to the CPU **165**. Thus the system bus **164** couples the CPU not only to the peripheral devices that are built into the portable computer **160**, but also the added components for the multimedia interface. The peripheral devices built into the portable computer **160** includes a display driver **166**, hard disk controller **168**, keyboard driver or controller **170**, a two-megabyte RAM **172**, ROM **174** for storing the system's basic input/output control software, and serial and parallel ports **176**.

The peripheral devices for the multimedia interface that are coupled to the system bus **164** includes a video display controller **180**, a controller **182** for the video image storage device **184**, a voice recognition module **186**, and a speech production module **188**.

The four add-on boards used by the multimedia interface are all compatible with the computer's system bus, and occupy the portions of the systems RAM address space, I/O address space, and interrupt vector shown in Table 4.

TABLE 4

Module	I/O Address	RAM Address	Int
Speech Production	300–30F		4
Voice Recognition		CE000 - CFFFF	5
Video Disk Controller		D8000 - DFFFF	3
Video Display Cntrlr	380 - 38F	A0000 - AFFFF	

The system also has 640K of RAM occupying addresses 00000–9FFFF for program execution, plus approximately 1.5 megabytes of additional RAM occupying addresses 10000– 27FFFFF which is usable as extended RAM or a virtual disk for speeding up the system, if needed.

The microcomputer's display **58** is a gas plasma text display. This display is separate from the video monitor **86**, discussed below. The text display is 10 inches wide by 8 inches in height, with a resolution of 64 0 by 400 pixels, and supports both upper and lower-case characters. The screen's size and resolution are more than sufficient to enable the average user to read the display from three to four feet away.

The microcomputer's standard keyboard is replaced with the keyboard **55** shown in FIGS. **5**A and **5**B. The keys in the keyboard **55** are scanned using the computer's standard keyboard controller **170**.

A compact video display monitor **86** presents video images to the user, and can overlay images with graphics. The display, model VU **100** made by Hycom, is 5 inches, measured diagonally, handles 4-MHz video bandwidth, and produces **16** shades of gray. The speed of the display is adequate to show multiframe video "movies" or sequences without noticeable flicker or blurring. Since detail information in both video images and schematic displays is of primary concern, the use of a color display was not considered to be important. Future embodiments may use a color video monitor if compact color monitors with sufficient resolution become available.

In the preferred embodiment, video images are stored in a WORM (write once, read many) disk memory **184** made by Maxtor. It is noted that the Maxtor disk memory **184** uses replaceable disks. In an alternate embodiment, the replaceable disks for the Maxtor disk memory could be used to store not only video images, but also a knowledge base and/or application programs which would be copied to the computer's hard disk 162 —thereby providing a convenient mechanism for loading new knowledge bases and application software into the system. In another alternate embodiment, a CD-I ROM could be used to store both digital data (i.e., knowledge base data and/or application software) and video information.

For voice output, a text-to-speech synthesizer **160**, the Prose **4000** made by Speech Plus, is used to minimize message storage requirements and to make it easier to implement new knowledge bases. Words requiring customized pronunciation can be added to a custom pronunciation dictionary, enabling the text-to-speech synthesizer to enunciate a virtually unlimited vocabulary. The voice output produced by the synthesizer **160** is used to drive both a loudspeaker and also the speakers in a headset **190** worn by the user.

The printer **90**, a Seiko model DPU-**411**, is an integral part of the system and can be accessed through a cover (not shown) behind the video and text displays for changing paper. This printer uses 4 inch wide, thermal paper with a print width of 40 columns and has print quality typical for a dot matrix. The printer's built-in interface (not shown), coupled to the computer's parallel port **172**, supports graphics and upper- and lower-case text. In future embodiments, a nonthermal full width printer would be preferred, if one of sufficient ruggedness, compact overall size, and weight becomes commercially available. Power for the system is provided by a standard line current at 120 VAC. The connection is made through a rugged industrial standard plug-in connector, fifteen feet in length. By using standard line current for power, the system can be easily used in virtually all locations without depending on a short life rechargeable battery.

Portability Requirements.

As described above, the system is designed to be hand-carried by an individual to various work sites. Therefore the system can be neither larger nor heavier than can be easily carried by one person.

The prototype shown in FIG. **1** is housed in a suitcase-like carrying case **200** that has a volume of 1.7 cubic feet. A card cage (not shown) below the portable computer couples the boards **202** and video disk **204** for the multimedia interface to the computer's system bus. The entire prototype weighs approximately 35 pounds.

As shown in FIG. **1**, the video and text displays **86** and **58** are mounted on hinges allowing the displays to be folded against the keyboard **55**, which enables the entire system to fit inside a reasonably small housing or carrying case. The text display **58** which comes with the portable computer incorporated in the system is already hinge mounted. The video display is mounted in a similar fashion and is coupled to the text display so that the two displays fold up and down in unison. As shown, the carrying case **200** includes a base member **206** which holds the computer and its peripheral equipment, and a hinged cover member **208** which closes over the entire system when the displays are folded down to provide a sealed, self-contained package.

It is anticipated that the production model, using a lighter weight frame and certain lighter components, will weigh between 25 and 30 pounds. With its carrying case **200** closed and locked, the system is able to withstand ambient weather conditions for a reasonable period of time and is essentially immune to vibrational damage resulting from vehicular transportation.

Video Image Sequences and Text for Equipment Maintenance Tasks

As shown in FIG. **1**, the video and text displays are juxtaposed to allow side by side presentation of video and text displays.

Referring to FIG. **8**, there is shown a side by side presentation of a video image and corresponding text generated by the preferred embodiment of the invention described above.

One of the original goals and anticipated uses of the present invention is to provide a practical equipment maintenance assistant. To provide practical assistance, the system must be able to show video images of the various pieces of equipment that may need to be repaired, and must also display text explaining how to perform various maintenance and/or repair tasks. In industrial settings, the system must also provide video images showing the locations of various pieces of equipment at a particular site, or at least the typical relative locations of various pieces of equipment in a system.

The preferred embodiment provides all of these capabilities. The expert system's knowledge base includes knowledge regarding the operation and maintenance of a predefined set of equipment, such as the equipment in a particular power plant and the equipment related to a particular set of turbine generators. The knowledge base includes references to (i.e., multimedia commands for) video images, and text to be displayed when the video images are displayed. In addition, the knowledge

base includes text to be spoken when corresponding video images and/or text strings are displayed, and additional text to be spoken for users who need or request more detailed explanations.

The expert system software prompts the user, through a series of questions, to identify a particular maintenance task. In some situations, the system identifies a maintenance task based on a set of symptoms related to the performance of various pieces of equipment.

When a particular piece of equipment has been identified, and/or a particular maintenance task has been specified, the system selects and displays one or more video images corresponding to the selected piece of equipment or specified maintenance task. The video images generated can be either a single video image or a specified sequence of images such as a video showing someone perform a typical repair job. At the same time that the video images are being displayed, the system displays text corresponding to those video images, e.g., text related to a specified equipment maintenance task.

In the preferred embodiment, the system also speaks aloud text corresponding to the displayed video images and displayed text, and will speak aloud additional text when the user requests a more detailed explanation.

By providing all of these capabilities in a portable housing, the present invention enables a computer assistant to be taken to and used at the locations where various maintenance tasks must be performed.

Alternate Embodiments

It should be understood that from the broadest perspective the present invention is not an expert system. It is a multimedia interface usable with many types of expert systems as well as other application software. Thus one set of alternate embodiments of the present invention is derived from the fact that the invention is usable with a variety of expert systems.

The most ambitious embodiment of the present invention currently being developed is an expert system called the START UP ADVISOR, which diagnoses a wide range of start up failures on gas turbines used for electrical power generation. While this expert system's knowledge base will be much more extensive than the one in the preferred embodiment described above, the multimedia interface in this embodiment of the invention will be virtually identical to the one in the preferred embodiment.

As will be understood by those skilled in the art, in other embodiments of the invention the multimedia commands generated by the expert system do not need to be "embedded" in the text strings which are to be displayed on the system's text monitor. The method of embedding multimedia commands in the text was devised so as to minimize the number of changes that would need to be made to an existing expert system in order to incorporate the present invention.

When designing a new expert system which will include the multimedia features of the present invention, the "output parser" can be dispensed with simply by providing separate storage for the display text and the multimedia commands. In the simplest such example, display text would be stored in one array, and all multimedia commands would be stored in another array. When displaying a selected set of text on the system's text monitor, the corresponding multimedia commands would set to a queue. The only difference from the above described preferred embodiment is that there is no need to "parse" the multimedia commands from the output generated by the inference software.

In another example, using the well known "code book" model for accessing a large database, when the expert system's inference software selects a portion of the knowledge base, the software will actually select a set of reference or code book pointers. These pointers will directly or indirectly denote display text, video displays, text for voice generation, and so on. Using these pointers the appropriate text can be directed to the system's text monitor while other sets of information are sent to the system's multimedia queue.

As noted earlier, the present invention can be used with expert systems which do not work along the lines of the simplified expert system model described above. From a more general perspective, the knowledge base in an expert system stores a specified set of information. The exact manner in which that information is organized is not particularly important for the purposes of the present invention. What is important from the perspective of the present invention is that the inference software in the expert system accesses selected portions of the information in the knowledge base, in accordance with the user's commands

Using the selected information, the expert system displays text on a text monitor and generates multimedia commands denoting additional information that can be presented by a multimedia interface when the corresponding text is displayed. The method of storing and/or generating these multimedia commands may vary widely from system to system. In accordance with the present invention, however, at least some of the multimedia commands must be queued, and the system must also have a dispatcher for executing the queued multimedia commands. Generally, the system will automatically generate certain predefined dispatch commands when displaying corresponding text, and will generate other dispatch commands at the user's request.

From an even broader perspective, the invention can be used with a wide range of application programs. To work with the present invention, the application program will generate multimedia commands denoting information that can be presented to the user by a multimedia interface. In addition, the system must queue at least some of the multimedia commands, and provide a dispatcher for executing the queued multimedia commands.

Another set of alternate embodiments of the present invention will inevitably arise as the hardware available for each of the input and output media improve over time, thereby improving the portability and usefulness of the invention. Even during the time during which the preferred embodiment was developed, lower weight and higher quality components became available and were used to replace previously selected components. Along these lines, it is entirely predictable that the weight and quality of portable video monitors and text monitors will improve over time. In fact, it is possible that it will become practical to use a single monitor for both text and video images, with the text of video images occupying different portions of a split screen.

In another example, it is likely that both voice input and speech generation technologies will improve over time, increasing the range of voice input commands that it is practical to use, and also increasing the usefulness of generated speech.

While the present invention has been described with reference to a few specific embodiments, the descrip-

tion is illustrative of the invention and is not to be construed as limiting the invention. Various modifications may occur to those skilled in the art without departing from the true spirit and scope of the invention as defined by the appended claims.

What is claimed is:

1. In a computer system, a multimedia interface comprising:

input means for entering user commands;

multimedia output means for presenting information to the user of said system using a plurality of output media; said multimedia output means including text displaying means for displaying text strings, and at least one additional output media;

an application program coupled to said input means and multimedia output means, said application program including means for generating output strings, said output strings including text strings for immediate delivery to a user of said computer system via said text display means, a multiplicity of said output strings further including means for denoting multimedia commands denoting information that can be presented by said multimedia output means;

queue means for receiving and storing said multimedia commands; and

dispatch means coupled to said queue means for executing said multimedia commands stored in said queue means by delivering said multimedia commands to said multimedia output means and thereby presenting said information denoted by said multimedia commands via said multimedia output means.

2. A computer system as set forth in claim 1,

said system including output processing means for processing output strings generated by said application program by storing the multimedia commands, if any, in said output strings in said queue means, delivering the text strings, if any, in said output strings to said text displaying means, and then activating said dispatch means to deliver said multimedia commands stored in said queue means to said at least one additional output media.

3. A computer system as set forth in claim 1, said multimedia output means including voice generating means for generating spoken messages;

said multimedia commands including a first set of voice output commands for denoting a first set of voice messages, and a second set of voice output command for denoting a second distinct set of voice messages;

said queue means including buffer means for storing specified voice messages;

said dispatch means including means for delivering said first set of voice commands to said voice generating means so as to speak said first set of voice messages, and for storing said second set of voice commands in said buffer means;

said system further including means, responsive to a predefined user command entered through said input means, for delivering said voice commands stored in said buffer means to said voice generating means so as to speak said stored voice messages.

4. A computer system as set forth in claim 1,

said multimedia output means including a second output means distinct from said text displaying means for communicating specified sets of information;

said multimedia commands including first and second sets of output commands, denoting first and second sets of information to be communicating by said second output means;

said queue means including buffer means for storing specified sets of information;

said dispatch means including means for delivering said first set of output commands to said second output means so as to communicate said first set of information, and for storing said sets of information denoted by second set of output commands in said buffer means;

said system further including means, responsive to a predefined user command entered through said input means, for delivering said sets of information stored in said buffer means to said second output means so as to communicate said stored sets of information.

5. An expert system as set forth in claim 1, said multimedia output means including means, coupled to said input means, for suspending the presentation of information to the user of said system when said input means receives a user command.

6. An expert system as set forth in claim 5, said input means including keyboard input means for receiving user commands and voice recognition means for receiving spoken user commands.

7. An expert computer system, comprising:

input means for entering user commands;

text display means for displaying text messages;

multimedia output means for presenting information to the user of said system using a plurality of output media in addition to said text display means;

a knowledge base storing a specified set of information, including means for denoting text strings that can be displayed on the text display means and corresponding multimedia commands denoting additional information that can be presented by said multimedia output means;

queue means for receiving and storing selected ones of said multimedia commands;

inference software means, coupled to said input means, text display means, knowledge base and queue means, for accessing selected portions of the information in said knowledge base, responding to user commands from said input means, sending text strings corresponding to selected portions of said information to said text display means for display, and storing in said queue means selected multimedia commands corresponding to said selected portions of said information in said knowledge base; and

dispatch means coupled to said queue means for delivering said multimedia commands stored in said queue means to said multimedia output means in response to a predefined dispatch command;

wherein said inference software includes means for generating said predefined dispatch command at predefined times after accessing selected portions of the information in said knowledge base.

8. An expert system as set forth in claim 7, said inference software including means for automatically generating said predefined dispatch command after sending said text strings corresponding to said accessed information to said text display means for display.

9. An expert system as set forth in claim 7,

said multimedia output means including voice generating means for generating spoken messages;

said multimedia commands including a first set of voice output commands for denoting a first set of voice messages, and a second set of voice output command for denoting a distinct second set of voice messages;

said queue means including buffer means for storing specified voice messages;

said dispatch means including means for delivering said first set of voice commands to said voice generating means so as to speak said first set of voice messages, and for storing said second set of voice commands in said buffer means;

said system further including means, responsive to a predefined user command entered through said input means, for delivering said voice commands stored in said buffer means to said voice generating means so as to speak said stored voice messages.

10. An expert system as set forth in claim **7**,

said multimedia commands including first and second sets of output commands, each denoting a distinct set of messages to be generated by said multimedia output means;

said system including means for automatically delivering said first set of output commands to said multimedia output means to present the information denoted by said first set of output commands after said inference software means sends text strings corresponding to selected portions of said information in said knowledge base to said text display means for display; and means responsive to a predefined user command entered through said input means, for delivering said second set of output commands to said multimedia output means to present the information denoted by said second set of output commands

11. An expert system as set forth in claim **7**, said multimedia output means including means, coupled to said input means, for suspending the presentation of information to the user of said system when said input means receives a user command.

12. An expert system as set forth in claim **11**, said input means including keyboard input means for receiving user commands and voice recognition means for receiving spoken user commands.

13. In an expert system having a knowledge base and inference software which accesses selected portions of the knowledge base, input means for user entry of commands, text display means for displaying text messages, the knowledge base including means for storing a multiplicity of text strings that can be displayed on the text display means, the inference software including means for responding to user commands from the input means and for displaying selected ones of said text strings stored in the knowledge base; the combination comprising:

multimedia output means for presenting information to the user of said system using a plurality of output media in addition to said text display means;

means for storing multimedia commands in the knowledge base along with corresponding ones of said text strings; said multimedia commands being embedded in said corresponding text strings;

queue means for storing multimedia commands;

output parsing means for removing and storing in said queue means said multimedia commands embedded in said text strings when said inference software selects corresponding portions of said knowledge base; and

dispatch means coupled to said queue means for executing said multimedia commands stored in said queue means by delivering said multimedia commands to said multimedia output means and thereby presenting said information denoted by said multimedia commands via said multimedia output means;

wherein said inference software includes means for generating a predefined dispatch command at predefined times after accessing a selected portion of the knowledge base.

14. An expert system as set forth in claim **13**, said multimedia output means including voice output means for converting a specified text string into synthesized speech;

said means for storing multimedia commands in the knowledge base including means for storing speech production commands in the knowledge base, said speech production commands including text strings representing words to be spoken when corresponding portions of the knowledge base are accessed by said inference software.

15. An expert system as set forth in claim **13**, said multimedia output means including voice output means for converting a specified text string into synthesized speech;

said multimedia commands including a first set of voice output commands for denoting a first set of voice messages, and a second set of voice output command for denoting a distinct second set of voice messages; said voice messages comprising text strings representing words to be spoken by said voice output means;

said queue means including buffer means for storing specified voice messages;

said dispatch means including means for delivering said first set of voice commands to said voice output means so as to speak said first set of voice messages, and for storing said second set of voice commands in said buffer means;

said system further including means, responsive to a predefined user command entered through said input means, for delivering said voice commands stored in said buffer means to said voice output means so as to speak said stored voice messages.

16. A method of processing and presenting information in a computer system, said computer system including text display means for displaying text messages; the steps of the method comprising:

providing multimedia output means for presenting information to the user of said system using a plurality of output media in addition to said text display means;

running an application program on said computer system said application program generating text strings to be displayed on said text display means and corresponding multimedia commands denoting additional information that can be presented by said multimedia output means;

displaying said text strings generated by said application program, and storing said multimedia commands generated by said application program in a queue; and

delivering said stored multimedia commands to said multimedia output means after said text string displaying step.

17. The method set forth in claim **16**,

said multimedia output means including voice generating means for generating spoken messages;

said multimedia commands including a first set of voice output commands for denoting a first set of voice messages, and a second set of voice output command for denoting a second distinct set of voice messages;

said method including the steps of:

providing a voice message buffer for storing voice messages;

said method including the step of delivering said first set of voice commands to said voice generating means so as to speak said first set of voice messages, and for storing said second set of voice commands in said voice message buffer;

receiving user commands;

responding to a predefined user command by delivering said voice commands stored in said voice message buffer to said voice generating means so as to speak said stored voice messages.

18. The method set forth in claim **17**, said step of receiving user commands including the step of suspending the operation of said voice generating means when a user command is received, thereby interrupting the speaking of said voice messages by said voice generating means.

19. A method of processing and presenting information in an expert computer system, said expert computer system including text display means for displaying text messages; the steps of the method comprising:

providing multimedia output means for presenting information to the user of said system using a plurality of output media in addition to said text display means;

providing a knowledge base storing a multiplicity of information records, each information record including means for denoting text strings that can be displayed on the text display means and corresponding multimedia commands denoting additional information that can be presented by said multimedia output means;

receiving user commands;

accessing selected information records in said knowledge base in response to said user commands;

displaying said text strings denoted by said accessed information records on said text display means, and storing said multimedia commands denoted by said accessed information records in a queue; and

delivering said stored multimedia commands to said multimedia output means after said text string displaying step.

20. A method of processing and presenting information in an expert computer system, said expert computer system including text display means for displaying text messages; the steps of the method comprising:

providing multimedia output means, including video display means for displaying video images and voice generating means for generating spoken messages;

providing a knowledge base storing a specified set of information including text strings that can be displayed on the text display means, and corresponding multimedia commands denoting video images and voice messages that can be presented by said video display means and said voice generating means;

receiving user commands;

accessing selected portions of the information in said knowledge base in response to said user commands;

displaying on said display means text strings corresponding to said selected portions of the information in said knowledge base, and queuing corresponding ones of said multimedia commands denoting video images and voice messages stored in said knowledge base; said multimedia commands including distinct first and second sets of commands;

delivering said first set of queued multimedia commands to said multimedia output means after displaying corresponding text strings; and

delivering said second set of queued multimedia commands to said multimedia output means when a corresponding user command is received

21. A method of processing and presenting information in an expert computer system, said expert computer system including text display means for displaying test messages; the steps of the method comprising:

providing multimedia output means, including video display means for displaying video images and voice generating means for generating spoken messages;

providing a knowledge base storing a specified set of information related to a predefined set of tasks to be performed on a predefined set of equipment; said stored information including text strings, corresponding video images showing said set of equipment, and corresponding voice messages;

receiving user commands;

accessing selected portions of the information in said knowledge base in response to said user commands, including selected text strings, and corresponding video images and voice messages; and

then displaying said selected text strings on said text display means, displaying corresponding video images on said video image means, and generating at least a portion of said corresponding selected voice messages with said voice generating means.

22. The method set forth in claim **21**,

said corresponding selected voice messages including first and second sets of said voice messages;

said generating voice messages step including the step of generating said first set of corresponding selected voice messages with said voice generating means;

said method further including the step of generating said second set of corresponding selected voice messages with said voice generating means when a predefined user command is received.

23. A method of processing and presenting information in an expert computer system, said expert computer system including text display means for displaying text messages; the steps of the method comprising:

providing multimedia output means, including video display means for displaying video images and voice generating means for generating spoken messages;

providing a knowledge base storing a specified set of information including text strings that can be displayed on the text display means, and corresponding multimedia commands denoting video images and voice messages that can be presented by said video display means and said voice generating means;

said knowledge base storing information related to a predefined set of tasks to be performed on a prede-

fined set of pieces of equipment; said video images denoted by multimedia commands including a multiplicity of images showing said set of pieces of equipment, including images showing said pieces of equipment in the context of said predefined set of tasks; said information stored by said knowledge base including text strings and voice messages related to said predefined set of tasks;

receiving user commands;

accessing selected portions of the information in said knowledge base in response to said user commands;

displaying on said display means text strings corresponding to said selected portions of the information in said knowledge base, and queuing corresponding multimedia commands denoting video images and voice messages stored in said knowledge base; said multimedia commands including distinct first and second sets of commands;

dispatching said first set of queued multimedia commands to said video display means and voice generating means at predefined times after displaying corresponding text strings; said dispatching step causing said expert system to simultaneously display text strings and video images related to selected ones of said predefined set of tasks;

and dispatching said second set of queued multimedia commands to said video display means and voice generating means when a corresponding user command is received; said second dispatching step causing said expert system to provide additional explanatory material related to selected ones of said predefined set of tasks.

24. A method of processing and presenting information in an computer system, said computer system including text display means for displaying text messages; the steps of the method comprising:

providing multimedia output means, including video display means for displaying video images and voice generating means for generating spoken messages;

providing an application program which generates output strings, said output strings including text strings for display on said text display means, a multiplicity of said output strings including multimedia commands denoting video images and voice messages that can be presented by said video display means and said voice generating means; said multimedia commands including distinct first and second sets of commands;

receiving user commands;

displaying said text strings on said text display means;

queuing said multimedia commands generated by said application program; and

dispatching said first set of queued multimedia commands to said multimedia output means after displaying corresponding text strings; and

dispatching said second set of queued multimedia commands to said multimedia output means when a corresponding user command is received.

25. In a computer system, the combination comprising:

selecting means for selecting a piece of equipment from a predefined set of pieces of equipment; said selecting means including voice recognition means for receiving spoken user commands;

video means for storing and displaying digitized video images corresponding to said selected piece of equipment, including video random access memory means for storing a multiplicity of digitized video images, each said stored image corresponding to one or more of said pieces of equipment and display means coupled to said video random access memory means for accessing and displaying at least one digitized video image stored in said video random access memory corresponding to said selected piece of equipment; and

output means including text displaying means for displaying text corresponding to said displayed video images, and voice generating means for generating spoken messages corresponding to said display video images;

whereby said computer system can be used to provide video and textual displays as well as spoken messages when working with a selected piece of equipment.

26. The computer system set forth in claim **25**, wherein said video means includes means for displaying a sequence of video images corresponding to said selected piece of equipment.

27. The computer system set forth in claim **26**, wherein said video means includes means for displaying a sequence of video images showing the location of said selected piece of equipment at a particular predefined site.

28. The computer system set forth in claim **27**, including task specifying means for specifying an equipment maintenance task to be performed;

said video means including means for displaying a sequence of video images related to said specified equipment maintenance task; and

said text means includes means for displaying text related to said specified equipment maintenance task.

29. The computer system set forth in claim **28**, including

portable housing which houses said computer system; said portable housing enabling said computer system to be taken to and used at the place where said selected piece of equipment is located.

30. The computer system set forth in claim **25**, wherein said video means includes

digitized plant layout images showing the locations of at least a multiplicity of said predefined set of pieces of equipment at a particular predefined site stored in said video random access memory means; and

video selecting means coupled to said video random access memory means and said video display means for displaying selected ones of said plant layout images corresponding to said selected piece of equipment.

31. The computer system set forth in claim **25**, including

portable housing which houses said computer system; said portable housing enabling said computer system to be taken to and used at the place where said selected piece of equipment is located.

32. In a computer system, the combination comprising:

means for defining a predefined set of pieces of equipment and a predefined set of maintenance tasks in conjunction with corresponding ones of said pieces of equipment;

task specifying means for specifying one of said equipment maintenance tasks to be performed in conjunction with corresponding ones of said pieces

of equipment; said task specifying means including voice recognition means for receiving spoken user commands;

video means for storing and displaying one or more digitized video images corresponding to said selected piece of equipment, including video random access memory means for storing a multiplicity of digitized video images, each said stored image corresponding to one or more of said pieces of equipment, and display means for displaying a sequence of said stored digitized video images related to said specified equipment maintenance task; and

text means for simultaneously displaying text corresponding to said video images, including means for displaying text related to said specified equipment maintenance task;

whereby said computer system can be used to provide assistance in the form of video sequences and textual displays to a person performing a specified equipment maintenance task.

33. The computer system set forth in claim **32**, including

portable housing which houses said computer system; said portable housing enabling said computer system to be taken to and used at the place where said selected piece of equipment is located.

34. In a computer system, a method of obtaining information regarding various pieces of equipment, the steps of the method comprising:

storing a multiplicity of digitized video images, each said stored image corresponding to one or more of said pieces of equipment from a predefined set of pieces of equipment;

selecting a piece of equipment from said predefined set of pieces of equipment; said selecting step including the step of receiving spoken user commands;

displaying at least one digitized video image stored in said video random access memory corresponding to said selected piece of equipment;

displaying text corresponding to said displayed video images, and

generating spoken messages corresponding to said displayed video images;

whereby a person using said computer system receives information in the form of digitized video images, textual displays and spoken messages while working with a selected piece of equipment.

35. The method obtaining information regarding various pieces of equipment set forth in claim **34**, wherein said displaying step includes displaying a sequence of said digitized video image corresponding to said selected piece of equipment.

* * * * *